MP-A40 Series
SDK for iOS
Application Programmer's Guide

U00136276710

Seiko Instruments Inc.

U00136276700	August 2016
U00136276701	October 2016
U00136276702	January 2018
U00136276703	February 2019
U00136276704	November 2019
U00136276705	September 2020
U00136276706	July 2022
U00136276707	April 2023
U00136276708	July 2024
U00136276709	December 2024
U00136276710	June 2026

Copyright © 2016-2026 by Seiko Instruments Inc. All rights reserved.

App Store℠ is a service mark of Apple Inc.
iPad®, iPad Air®, iPad mini®, iPhone®, and iPod® are trademarks of Apple Inc., registered in the U.S. and other countries and regions.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.
IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.
All other trademarks are the properties of their respective companies.

Seiko Instruments Inc. (hereinafter referred to as ”SII”) has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

SII is a trademark of Seiko Instruments Inc.

Introduction

This document describes the MP-A40 Series SDK for iOS (hereinafter referred to as "the SDK") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Printers

This section lists the printers supported by the SDK.

Printer	Communication Interface
MP-A40 (Bluetooth supporting model)	Bluetooth
MP-A40 (Wireless LAN supporting model)	Wireless LAN

Table of Contents

Chapter 1 Product Overview 1-1

1.1 Function 1-1
1.2 Configuration 1-1
1.2.1 SII Print Class Library 1-2
1.2.2 Sample Program 1-2

Chapter 2 Product Specifications 2-1

2.1 Operating Environment 2-1
2.1.1 Applicable iOS Devices 2-1
2.1.2 Applicable iOS Versions 2-2
2.2 Operating Conditions 2-2
2.3 Precaution 2-3

Chapter 3 How to Use Library 3-1

3.1 Developmental Environment for iOS Application 3-1
3.2 Provided Files 3-2
3.3 Build Library into Xcode Projects 3-3

Chapter 4 Function of Library 4-1

4.1 Overview of Library 4-1
4.2 Structure of Library 4-1
4.3 Receive Data Process and Limitations 4-1
4.4 API Reference 4-2
4.4.1 Class 4-4
(1) SIIMpaPrinterManager 4-4
init Instance 4-7
open Start using printer 4-7
close Finish using printer 4-8
setWriteTimeout Set send timeout period 4-9
getWriteTimeout Retrieve send timeout period 4-9
setResponseTimeout
Set receive timeout period 4-9
getResponseTimeout
Retrieve receive timeout period 4-10
write Send Binary data 4-10
read Retrieve receive data 4-10
getReadSize Retrieve available receive data size 4-11
writeAndWaitResponse
Send and receive binary data 4-11
reset Reset printer 4-12

getStatus Retrieve printer status....4-13
startCallbackFunction
Start callback of printer status change....4-13
stopCallbackFunction
Finish callback of printer status change....4-14
registerStyleSheet Register style sheet to printer....4-14
deleteStyleSheet Delete style sheet in printer....4-15
registerLogo Register logo to printer....4-15
deleteLogo Delete logo in printer....4-16
getPrinterInformation
getPrinterInformationNumber
getPrinterInformationString
Retrieve printer information....4-16
getVersion Retrieve SDK version....4-18
controlTransaction Start/finish print data batch sending....4-18
selectStandardMode
Select standard mode....4-19
selectPageMode Start page mode....4-20
printPageModeData
Print page mode data....4-20
setPageModeVerticalPosition
Specify vertical absolute position in page mode....4-21
setStandardModeArea
Specify print area in standard mode....4-21
setStandardModeAlignment
Alignment....4-22
setHorizontalPosition
Specify absolute position....4-23
setStandardModeBarcodeDirection
Select print direction for barcode or 2-dimensional barcode....4-23
setLineSpacing Specify line space amount....4-24
setCharacterRightSpace
Specify character right space amount....4-24
selectCharacterSet Select character set....4-25
getCharacterSet Retrieve specified character set....4-25
selectInternationalCharacterSet
Select international character set....4-25
getInternationalCharacter
Retrieve specified international character set....4-26
setCharacterFormatting
Format character....4-26
printText Send text data....4-27
printLogo Print logo registered in printer....4-27
setImageTransactionMode
Specify image printing mode....4-28
getImageTransactionMode
Retrieve image printing mode....4-28
sendDataFile Send file data....4-28

sendImage Send image 4-29
printPDF Print PDF page 4-30
printBarcode Print barcode 4-32
print2DCode Print 2-dimensional barcode 4-36
printPageModeRectangle
Draw rectangular 4-38
feedLine Feed paper by line 4-38
feedDotLine Feed paper by dot 4-39
feedCutPosition Feed paper to cut position 4-39
feedMarkPosition Marked paper form feed 4-40
setLog Specify log output 4-40
isOpened Retrieve printer using status 4-41
(2) SIIMpaPrinterStatus 4-42
errOffline Retrieve offline error status 4-42
errHardware Retrieve hardware error status 4-42
errVoltage Retrieve Vp Voltage error status 4-43
errHeadTemperature
Retrieve head temperature error status 4-43
errOutOfPaper Retrieve out-of-paper error status 4-43
errMarkPaperJam Retrieve paper jam error status while detecting marked paper. 4-43
errCoverOpen Retrieve paper cover open error status 4-43
errBattery Retrieve battery error status 4-43
stateFeedSwitch Retrieve feed switch status 4-44
statePaperFeed Retrieve paper feed status 4-44
stateReturnWaiting Retrieve return waiting status 4-44
stateFlashMemoryRewriting
Retrieve FLASH memory rewrite status 4-44
stateBattery Retrieve battery voltage status 4-44
(3) SIIMpaPrinterDiscovery 4-45
startDiscoveryPrinter
Start printer search 4-45
cancelDiscoveryPrinter
Cancel printer search 4-46
getFoundPrinter Retrieve information list of searched printer 4-47
(4) SIIMpaDiscoveredPrinter 4-48
printerModel Retrieve printer model name 4-48
bluetoothAddress Retrieve Bluetooth address 4-48
ipAddress Retrieve IP address 4-48
macAddress Retrieve MAC address 4-48
serialNumber Retrieve Serial Number 4-49
4.4.2 Enumerated Type (Enum) 4-50
(1) PrinterInterface 4-50
(2) PrinterModel 4-50
(3) Dithering 4-50
(4) PrinterInformation 4-50
(5) TransactionFunction 4-51
(6) Direction 4-52
(7) Alignment 4-52

(8) CharacterSet 4-53
(9) InternationalCharacterSet 4-54
(10) CharacterType 4-54
(11) CharacterScale 4-55
(12) Underline 4-55
(13) Bold 4-55
(14) Reverse 4-55
(15) Rotate 4-56
(16) TypeBarcode 4-57
(17) ModuleWidthBarcode 4-57
(18) HriPositionBarcode 4-58
(19) NwRatioBarcode 4-58
(20) Type2DCode 4-58
(21) Mode2Dcode 4-58
(22) ModuleSize2DCode 4-60
(23) ErrorCorrect2DCode 4-62
(24) LogFileSize 4-63
(25) ErrorCode 4-63
(26) ImageTransactionMode 4-63

Chapter 5 Sample Program 5-1

5.1 Sample Program Overview 5-1
5.2 Sample Program Function 5-2
5.2.1 API Window 5-2
5.2.2 Settings Window 5-4
5.3 Precaution 5-5

Chapter 6 Disclaimer 6-1

Appendix A Character Sets (Character Code Table) A-1

A.1 Character Code Table A-1
A.2 International Character Set A-11

Chapter 1
Product Overview

This chapter describes the product overview of the SDK.

1.1 Function

The SII print class library included in the SDK, provides iOS applications with the functions to use SII printer MP-A40 series (hereinafter referred to as "printer").
Moreover, the SDK provides Xcode projects as a sample program for SII print class library.

1.2 Configuration

The SII print class library and the sample programs included in the SDK are located in the section surrounded by dashed lines in the iOS configuration diagram (Figure 1-1).

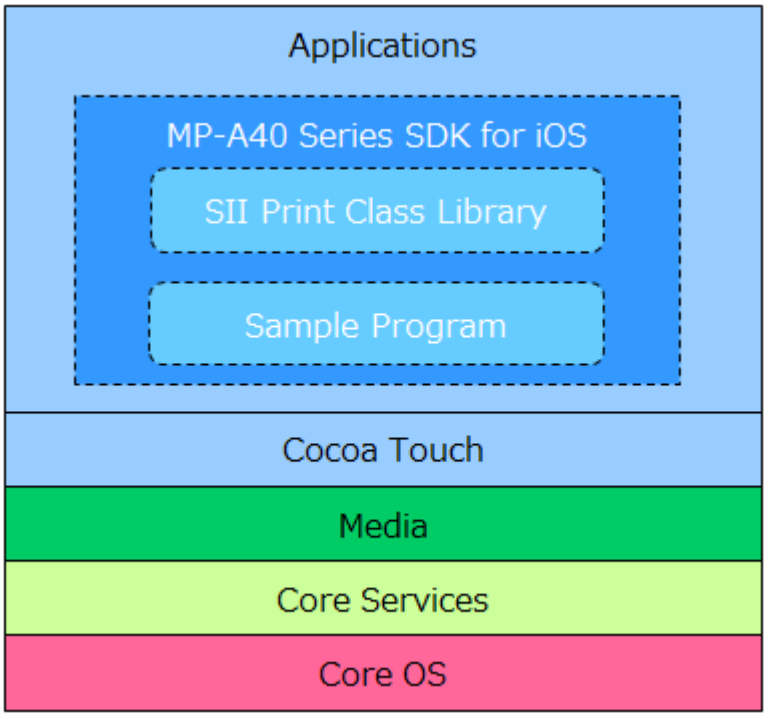


Figure 1-1

1.2.1 SII Print Class Library

By using SII Print Class Library (hereinafter referred to as "the library"), iOS applications can easily send print data and printer commands to printer through communication port (Bluetooth or TCP/IP) on an iOS device. Also, the applications can retrieve printer status.

The library provides the following functions.

- Connection/disconnection to/from a printer
- Sending data to a printer (print data and/or printer commands)
- Barcode print and 2-dimensional barcode print
- Sending a data file to a printer (print data and/or printer commands)
- Retrieving printer status
- Retrieving various responses from a printer
- Printer search by Bluetooth or TCP/IP

1.2.2 Sample Program

When an application used for communication with a printer by Bluetooth is registered to App Store, SII needs to apply it in advance. For the details, contact SII.

Chapter 2
Product Specification

This chapter describes the product specification of the library.

2.1 Operating Environment

2.1.1 Applicable iOS Devices

Applicable iOS devices for the library are shown in the following list.

iPhone models

- iPhone 16 Pro Max
- iPhone 16 Pro
- iPhone 16 Plus
- iPhone 16
- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone SE (2nd generation)

iPad models

- iPad mini (A17 Pro)
- iPad Pro 13-inch (M4)
- iPad Pro 11-inch (M4)
- iPad Air 13-inch (M2)
- iPad Air 11-inch (M2)
- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11-inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)

2.1.2 Applicable iOS Versions

Applicable iOS versions for the library are shown in the following list.

- iOS 17-17.6.1
- iPadOS 17-17.6.1
- iOS 18-18.6
- iPadOS 18-18.6
- iOS 26-26.5
- iPadOS 26-26.5

2.2 Operating Conditions

This section describes the operating conditions for the library in the following table.
Set the Function Setting / function selection to the values shown in the following table before using the library.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about Function Settings.

- **Memory Switch**

Table 2-1 Function Settings by Memory Switch

MS	Function	Value
1-1	Interface Selection (Interface)	1: USB/Wireless
1-2	Mark Mode Selection (Mark Mode)	0: Enable*1 1: Disable
1-3	Command System Selection (Command System)	000: ESC/POS
1-4		
1-5		
2-2	Realtime Command Selection (Realtime Command)	0: Enable
9-1	Automatic Status Response Selection <ESC/POS> (Auto Status Back <ESC/POS>)	0: Enable
9-2	Initialized Response Selection <ESC/POS> (Init. Response <ESC/POS>)	0: Enable

*1: Select "Enable" when feeding to home position of the marked thermal paper.

- **Bluetooth Communication Settings**

  For Bluetooth communication settings, set the following values in addition to Table 2-1.
  For the setting method, see the printer command "Set Bluetooth Communication" described in "MP-A40 SERIES Thermal Printer USER'S GUIDE".

Function	Value
iOS Auto Connection Selection (Auto Connection)	0: Disable 1: Enable*1
Profile Selection (Profile)	1: iAP2*2

*1: Select "Enable" when using **`reset`**.
*2: To switch the Profile Selection, press the POWER Switch 5 times within 3 seconds with the printer on.

2.3 Precaution

When using TCP/IP, the library cannot share the communication port with the printer driver or other libraries.
When using TCP/IP, wireless LAN access point which the iOS device is connected to and the printer need to be connected to the same network.

A concurrent connection from multiple apps to one printer is not supported when the multiple apps are started simultaneously by the Multitasking function for the iPad with iPadOS.

Chapter 3
How to Use Library

This chapter describes development environment for iOS application and how to use the library.

3.1 Developmental Environment for iOS Application

In order to develop iOS applications, following tools are required.

- Xcode 15.0 or later

This chapter and after in this document, it is assumed that there is an environment where you can use each tool.

3.2 Provided Files

The file configuration of the SII print class library is as follows.

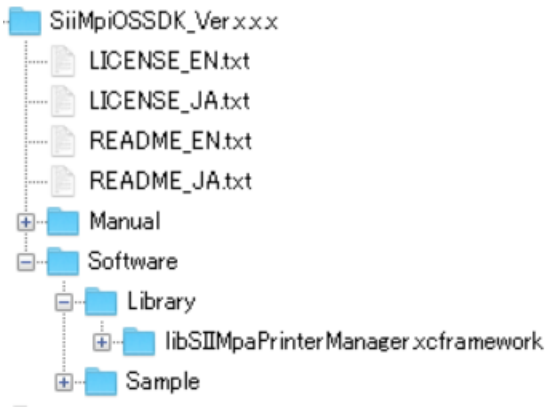


Figure 3-1

The file format of the library is XCFramework. The file name of the library is libSIIMpaPrinterManager.xcframework.

3.3 Build Library into Xcode Projects

Using the project of the sample program (SIIMpaPrinterSample) included in the SDK as an example, this section describes how to build the library into the projects.

See "Chapter 5 Sample Program" for sample programs included in the SDK.

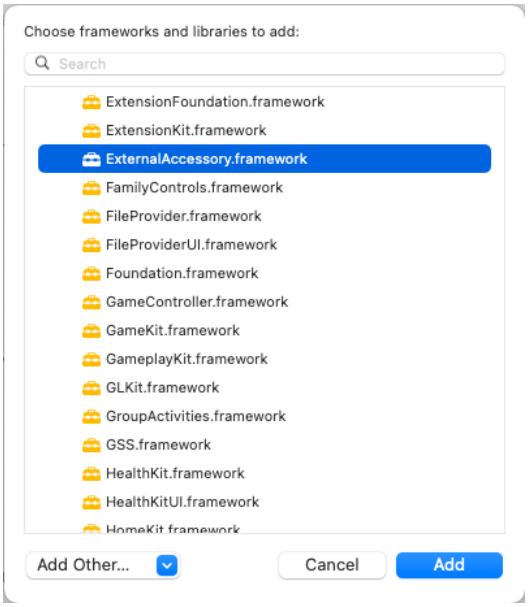
(NOTE) If the following libraries provided SDK Ver. 1.0.9 or earlier versions are included in the target project, delete them all.

- **SIIMpaPrinterManager.a**
- **SIIMpaPrinterManager.h**

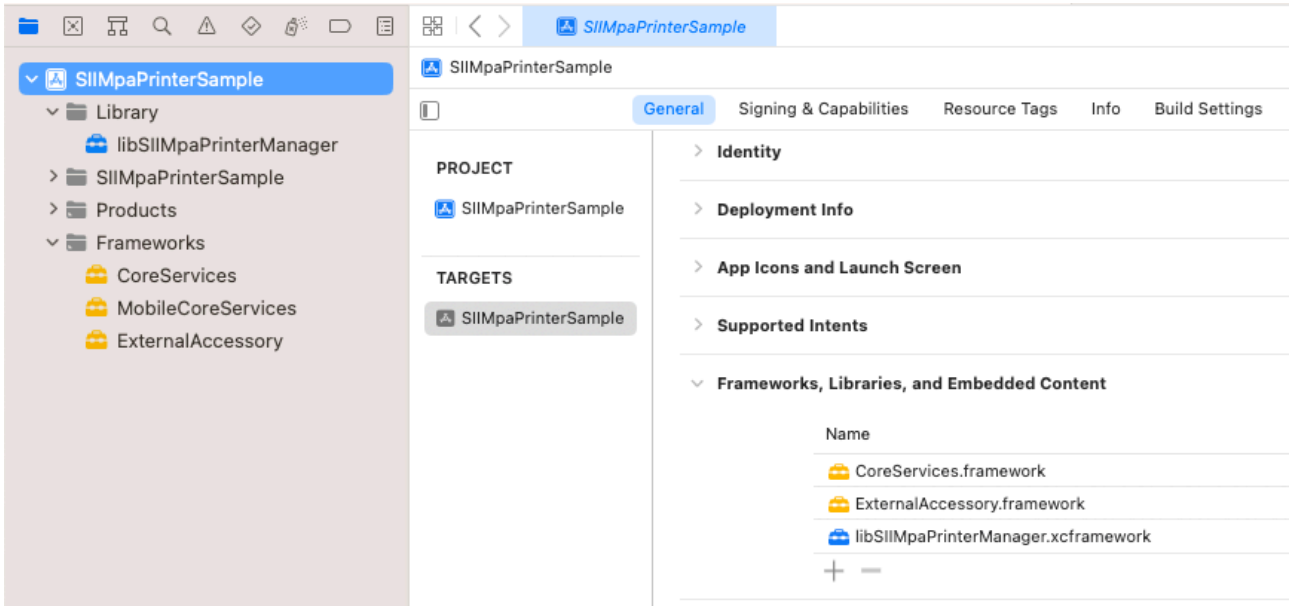
(1) Open the Xcode project.

(2) Build Framework and a library.

- Build ExternalAccessory.framework into the project.
  Select the target project in [Project Navigator], go [TARGETS] - [General] - [Frameworks, Libraries, and Embedded Content], open [Frameworks, Libraries, and Embedded Content].
  And then, click [+], select ExternalAccessory.framework from the list and click [Add].



- Build this library into the project.
  As well as the previous procedure, open [Frameworks, Libraries, and Embedded Content].
  And then, click [+] and select this library.



(3) Add the import definition for the built library to the source file of the application that uses this library as follows.

- For using Objective-C

```
#import <SIIMpaPrinterManager/SIIMpaPrinterManager.h>
```

- For using Swift

```
import SIIMpaPrinterManager
```

(4) Set the protocol name used by ExternalAccessory.framework.
Select xxxx.plist in [Project Navigator].
(For the sample program in the SDK, see SIIPrinterManagerSample-Info.plist.)

From the popup menu, select [Add Row] - [Supported external accessory protocols].

When open the added [Supported external accessory protocols], [Item 0] is displayed, then input "com.sii-ps.siieap" as the value.

▼ Supported external accessory protocols ⇕	Array	(1 item)
Item 0	String	com.sii-ps.siieap

By completing these procedures, functions of the library become available.

Chapter 4
Function of Library

This chapter describes the APIs for each class implemented in the library.

4.1 Overview of Library

The library provides the functions to use printers for iOS-enabled applications.

4.2 Structure of Library

The file format of the library is XCFramework. The file name of the library is libSIIMpaPrinterManager.xcframework.

In order to use the library in iOS applications, build the library into Xcode projects of iOS applications.

See "Chapter 3 How to Use the Library" for details about how to build the library into Xcode projects.

4.3 Receive Data Process and Limitations

This library buffers the receive data from the printer into the receive buffer in the SDK (hereinafter referred to as "the buffer") after starting to use the printer by **`open`**. Since the printer replies the auto status response every time the status changes, the receive data is buffered into the buffer sequentially. The maximum receive data to be buffered is 4096 bytes. When the printer is reconnected, the receive buffer in the SDK may collectively receive the data that was buffered in the printer before reconnecting.

The receive data buffered in the buffer can be retrieved by **`read`**. The receive data retrieved by **`read`** is deleted from the buffer.

When the buffered receive data exceeds 4096 bytes, the old data in buffer is discarded sequentially; therefore, execute **`read`** so that the receive data does not exceed 4096 bytes. The receive data size can be retrieved by **`getReadSize`**.

For the details of the each method, see "4.4 API Reference".

For the details of the each response, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

4.4 API Reference

- **Class**

Name	Description
SIIMpaPrinterManager	Class that provides the API used for communication with the printer and for printing.
SIIMpaPrinterStatus	Class that stores the printer status.
SIIMpaPrinterDiscovery	Class that searches the printer.
SIIMpaDiscoveredPrinter	Class that stores the printer information searched by printer searching.

- **Enumerated type (enum)**

Name	Description
PrinterInterface	Enumerator used for specifying the communication interface of **open**.
PrinterModel	Enumerator used for specifying the printer model of **open**.
Dithering	Enumerator used for dithering of **resisterLogo**, **sendDataFile**, and **printPDF**.
PrinterInformation	Enumerator used for specifying the printer information of **getPrinterInformation**, **getPrinterInformationNumber** and **getPrinterInformationString**.
TransactionFunction	Enumerator used for specifying the batch process control method of **controlTransaction**.
Direction	Enumerator used for specifying the print direction of **selectPageMode**.
Alignment	Enumerator used for specifying the print position of **setStandardModeAlignment**.
CharacterSet	Enumerator used for specifying the character set of **selectCharacterSet**.
InternationalCharacterSet	Enumerator used for specifying the international character set of **selectInternationalCharacterSet**.
CharacterType	Enumerator used for specifying the character font of **setCharacterFormatting**.
CharacterScale	Enumerator used for specifying the character scale of **setCharacterFormatting**.
Underline	Enumerator used for specifying the underline of **setCharacterFormatting**.
Bold	Enumerator used for specifying the bold print of **setCharacterFormatting**.
Reverse	Enumerator used for specifying the reverse print of **setCharacterFormatting**.
Rotate	Enumerator used for specifying the rotation print of **setCharacterFormatting**, for specifying print direction of **setStandardModeBarcodeDirection**, and for specifying the image rotation direction of **printPDF**.

Name	Description
TypeBarcode	Enumerator used for specifying the barcode type of **printBarcode**.
ModuleWidthBarcode	Enumerator used for specifying the barcode module width or narrow element of **printBarcode**.
HriPositionBarcode	Enumerator used for specifying the barcode HRI characters of **printBarcode**.
NwRatioBarcode	Enumerator used for specifying the barcode N:W ratio of **printBarcode**.
Type2DCode	Enumerator used for specifying the 2-dimensional barcode type of **print2DCode**.
Mode2Dcode	Enumerator used for specifying the 2-dimensional barcode mode of **print2DCode**.
ModuleSize2Dcode	Enumerator used for specifying the 2-dimensional barcode module size of **print2DCode**.
ErrorCorrect2Dcode	Enumerator used for specifying the 2-dimensional barcode error correction level of **print2DCode**.
LogFileSize	Enumerator used for specifying the maximum size of the log file of **setLog.**
ErrorCode	Enumerator used for retrieving the error code by **NSError**.
ImageTransactionMode	Enumerator used for specifying image printing mode in **setImageTransactionMode**.

4.4.1 Class

(1) `SIIMpaPrinterManager`

- **Summary**

This class provides the API used for communication with the printer and for printing.
This class provides the following functions.

Public Methods

Name	Description	Standard Mode[*1]	Page Mode[*1]
`SIIMpaPrinterManager`	Constructor	✓	✓
`init`	Instance	✓	✓
`open`	Start using printer	✓	✓
`close`	Finish using printer	✓	✓
`setWriteTimeout`	Set send timeout period	✓	✓
`getWriteTimeout`	Retrieve send timeout period	✓	✓
`setResponseTimeout`	Set receive timeout period	✓	✓
`getResponseTimeout`	Retrieve receive timeout period	✓	✓
`write`	Send binary data	✓	✓
`read`	Retrieve receive data	✓	✓
`getReadSize`	Retrieve available receive data size	✓	✓
`writeAndWaitResponse`	Send and receive binary data	✓	✓
`reset`	Reset printer	✓	✓
`getStatus`	Retrieve printer status	✓	✓
`startCallbackFunction`	Start callback of printer status change	✓	✓
`stopCallbackFunction`	Finish callback of printer status change	✓	✓
`registerStyleSheet`	Register style sheet to printer	✓	✓
`deleteStyleSheet`	Delete style sheet in printer	✓	✓
`registerLogo`	Register logo to printer	✓	✓
`deleteLogo`	Delete logo in printer	✓	✓
`getPrinterInformation`	Retrieve printer information	✓	✓
`getPrinterInformationNumber`		✓	✓
`getPrinterInformationString`		✓	✓
`gerVersion`	Retrieve SDK version	✓	✓
`controlTransaction`	Start/finish print data batch sending	✓	✓
`selectStandardMode`	Start standard mode	-	✓
`selectPageMode`	Start page mode	✓	-
`printPageModeData`	Print page mode data	-	✓[*2]
`setPageModeVerticalPosition`	Specify vertical absolute position in page mode	-	✓[*2]

Name	Description	Standard Mode*1	Page Mode*1
setStandardModeArea	Specify print area in standard mode	✓*3	-
setStandardModeAlignment	Alignment	✓*3	-
setHorizontalPosition	Specify absolute position	✓	✓
setStandardModeBarcodeDirection	Select print direction for barcode or 2-dimensional barcode	✓*3	-
setLineSpacing	Specify line space amount	✓*4	✓*4
setCharacterRightSpace	Specify character right space amount	✓*4	✓*4
selectCharacterSet	Select character set	✓	✓
getCharacterSet	Retrieve specified character set	✓	✓
setInternationalCharacterSet	Select international character set	✓	✓
getInternationalCharacter	Retrieve specified international character set	✓	✓
setCharacterFormatting	Format character	✓	✓
printText	Send text data	✓	✓
printLogo	Print logo registered in printer	✓	✓
setImageTransactionMode	Specify image printing mode	✓	✓
getImageTransactionMode	Retrieve image printing mode	✓	✓
sendDataFile	Send file data	✓	✓
sendImage	Send image data	✓	✓
printPDF	Print PDF page	✓*3	-
printBarcode	Print barcode	✓	✓
print2DCode	Print 2-dimensional barcode	✓	✓
printPageModeRectangle	Draw rectangular	-	✓*2
feedline	Feed paper by line	✓	✓
feedDotLine	Feed paper by dot	✓	✓
feedCutPosition	Feed paper to cut position	✓	✓
feedMarkPosition	Marked paper form feed	✓	✓
setLog	Specify log output	✓	✓

*1 ✓: Enable, -: Disable
*2 When use this method, execute **SelectPageMode** beforehand to start page mode.
*3 When use this method, execute **SelectStandardMode** beforehand to start standard mode.
*4 Independent settings are available for standard mode and page mode respectively.

Public Properties

Name	Description	Standard Mode*1	Page Mode*1
isOpened	Retrieve printer using status	✓	✓

*1 ✓: Enable, -: Disable

Constants

Constant Name	Description
SII_MPM_PAPER_WIDTH_80mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 80 mm Constant value = 576
SII_MPM_PAPER_WIDTH_100mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 100 mm Constant value = 736
SII_MPM_PAPER_WIDTH_105mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 105 mm Constant value = 776
SII_MPM_PAPER_WIDTH_112mm	Print width (dots) when the memory switches MS 3-1 to 6 (Paper Width Selection) of the printer is set to 112 mm Constant value = 832

- **Public Methods**

init — Instance

This method initializes the instance of **SIIMpaPrinterManager** class.

Syntax
```
- (id) init;
```

Return — When succeeded, the initialized instance of **SIIMpaPrinterManager** class is returned. When failed, **nil** is returned.

Example
```
SIIMpaPrinterManager *printerManager
                    = [[SIIMpaPrinterManager alloc]init];
```

open — Start using printer

Starts using a printer.

Syntax
```
- (BOOL) open: (PrinterInterface) printerInterface
             printerModel: (PrinterModel) printerModel
             address: (NSString*) address
             socketKeepingTime: (NSInteger) socketKeepingTime
             error: (NSError**)error;
```

Parameter

printerInterface	Communication interface constant See "4.4.2(1) **PrinterInterface**" for available settings.
printerModel	Printer model constant See "4.4.2(2) **PrinterModel**" for available settings.
address	Varies depending on **printerInterface** setting. ・When specify **SII_MPM_PRN_IF_TCP**: In **address**, specify IP address of the printer to connect. Example: "192.168.0.190", "host" ・When specify **SII_MPM_PRN_IF_BT**: In **address**, specify Bluetooth address of the printer paired with iOS device. Example: "00:11:22:AA:BB:CC"
socketKeepingTime	Varies depending on **printerInterface** setting. ・When specify **SII_MPM_PRN_IF_TCP**: The range is 60 to 300. When the specified value is below 60, the value is set to 60, when the specified value exceeds 300, the value is set to 300. In **socketKeepingTime**, specify the same value as the timeout period of unconnected communication specified in printer wireless LAN setting. The default value of the timeout period of unconnected communication is 300 seconds. For the details, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE". ・When specify **SII_MPM_PRN_IF_BT**: The specified value is ignored.
error	Error object reference.

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description This method connects to a printer specified by **printerInterface** through a communication interface specified by **printerModel**.

Monitoring the printer status is started by this method. The latest printer status is retrieved by **getStatus**. The change of printer status can be notified as an event by **onStatusChanged**, **startCallbackFunction**, or **stopCallbackFunction.**

Note Do not disable the automatic status response by printer command "Automatic Status Back Enable/Disable" or function settings. In that case, the printer status cannot be monitored, and the related function cannot be operated. For the automatic status response and the function settings, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When data is sent to a printer through TCP/IP connection and the socket keeping time is passed, communication socket is discarded. After that, a communication socket is created again and connection starts. If the printer is requested to connect from the other host on the same network at the timing of the communication socket discarding, the printer establishes the communication with the other host and the reconnection may be failed.

A concurrent connection from multiple apps to one printer is not supported.

close — Finish using printer

Finishes using the printer and monitoring the printer status.

Syntax `- (BOOL) close: (NSError**) error;`

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description This method stops communicating with the printer connected by **open**.
When this method is executed during data transmission, the communication will be disconnected after all data is transmitted and printing operation is completed, or after the receive timeout period set by **setResponseTimeout**.
Print data stored by **controlTransaction** is discarded by this method.

Note In Bluetooth communication, set sufficient receive timeout period in consideration of the time required for printing.
The receive timeout period needs approximately 10000 milliseconds for the print length 150 mm.
If Bluetooth communication is disconnected while sending data, the data transmission and reception may not be possible due to abnormal communication between the iOS device and the printer.

setWriteTimeout — Set send timeout period

Specifies the send timeout period.

Syntax

```
- (void) setWriteTimeout: (NSInteger) writeTimeout;
```

Parameter

writeTimeout	Send timeout period (millisecond) The range is 1000 to 300000. When the specified value is below 1000, the value is set to 1000. When the specified value exceeds 300000, the value is set to 300000. When the value is not set in this method, the send timeout period is 10000 seconds (default value).

Description

The specified value can be retrieved by **getWriteTimeout**.

This method is enabled in the following methods.

- **write**
- **writeAndWaitResponse** (Transmission process part)
- **registerLogo**
- **registerStyleSheet**
- **controlTransaction**
  (When **SII_MPM_TRANSACTION_PRINT** is selected in **control**)
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2Dcode**

getWriteTimeout — Retrieve send timeout period

Retrieves the send timeout period.

Syntax

```
- (NSInteger) getWriteTimeout;
```

Return value

Send timeout period (millisecond)

setResponseTimeout — Set receive timeout period

Specifies the receive timeout period.

Syntax

```
- (void) setResponseTimeout: (NSInteger) respTimeout;
```

Parameter

respTimeout	Receive timeout period (millisecond) The range is 1000 to 300000. When the specified value is below 1000, the value is set to 1000. When the specified value exceeds 300000, the value is set to 300000. When the value is not set in this method, the send timeout period is 10000 seconds (default value).

Description

The specified value is retrieved by **getResponseTimeout**.

This method is enabled in the following methods.

- **writeAndWaitResponse**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

getResponseTimeout — Retrieve receive timeout period

Retrieves the receive timeout period.

Syntax

```
- (NSInteger) getResponseTimeout;
```

Return value Receive timeout period (millisecond)

write — Send Binary data

Sends the binary data.

Syntax

```
- (BOOL) write: (NSData*) binary
               offset: (NSInteger) offset
               error: (NSError**) error;
```

Parameter

binary	Binary data to send to a printer The maximum data size that can be specified is 3 MB (3145728 bytes) .
offset	Specification of the starting position of the data send
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description When **controlTransaction** is not used, the timeout period specified in **setWriteTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use reset. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

read — Retrieve receive data

Retrieves the receive data which size is specified by **bufferSize** stored in **readBinary**.

Syntax

```
- (BOOL) read: (NSMutableData**) readBinary
              bufferSize: (NSInteger) bufferSize
              error: (NSError**) error;
```

Parameter

bufferSize	Receive data size (byte) The range is 1 to 4096. When the specified value exceeds 4096, the value is set to 4096. When the specified value is below 0, sends the error notice.

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description When entire receive data buffered in the buffer needs to be retrieved, specify the value in **bufferSize** retrieved by **getReadSize**.

When this method is reexecuted after retrieving receive data by this method, retrieving starts from the following receive data in the buffer.

For the procedure and the limitation, see "4.3 Receive Data Process and Limitations".

getReadSize — Retrieve available receive data size

Retrieves the available receive data size.

Syntax
```
- (BOOL) getReadSize: (NSInteger*) readSize
                error: (NSError**) error;
```

Parameter

readsize Receive data size (byte)

error Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

writeAndWaitResponse — Send and receive binary data

Sends the binary data and retrieves the receive data after this method is executed.

Syntax
```
- (BOOL) writeAndWaitResponse: (NSData*) sendBinary
                receiveBinary: (NSMutableData**) receiveBinary
                bufferSize: (NSInteger) bufferSize
                respControl: (BOOL)respControl
                incAsbData: (BOOL) incAsbData
                error: (NSError**) error;
```

Parameter

sendBinary Data to send
The maximum data size that can be specified is 3 MB (3145728 bytes).

reveiveBinary Data to receive

bufferSize Maximum size of the data to send
The range is 1 to 10485760.
When the specified value exceeds 10485760, the value is set to 10485760.

respControl Operation selection for receive process
YES Receive some data or continue to receive data until timeout period is over.
NO Receive the specified sized data in **bufferSize** or continue to receive data until timeout period is over.

incAsbData Include the automatic status response to the receive data or not

YES Include
Respond the receive data as it is.

NO Not include
Respond the receive data excluding the automatic status response.

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description This method is appropriate to send printer command for responding value of character string amount or capacity, and to retrieve the respond.

Next method execution after this method execution is not processed until completion of this method receiving. For retrieving the execution response or progress response, use **read** but not this method. For the details of execution response or progress response, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

For sending, the timeout period specified in **setWriteTimeout** is valid.

For receiving, the timeout period specified in **setResponseTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use reset. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

reset Reset printer

Resets the printer.

Syntax
```
- (BOOL) reset: (NSError**) error;
```

Parameter **error** Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description When this method is executed while the following method is being processed, the process is aborted. The unsent or unreceived data of following methods are discarded.

- **write**
- **writeAndWaitResponse**
- **controlTransaction**
  (When **SII_MPM_TRANSACTION_PRINT** is selected in **control)**
- **printText**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **getPrinterInformation**
- **getPrinterInformationNumber**
- **getPrinterInformationString**

While this method is being executed, the printer status becomes offline.

After this method is executed, wait for a few seconds before data send method is executed. Data send during reset may cause data lost.

When this method is executed but the printer is in the condition of no data accepting, the printer reset may not be executed normally, and there is a possibility of character corruption.

getStatus — Retrieve printer status

Stores the latest printer status to **getStatus**.

Syntax

```
- (BOOL) getStatus: (SIIMpaPrinterStatus**) getStatus
                error: (NSError**) error;
```

Paramater

getStatus	Printer status
error	Error object reference

Return value

Printer status is returned by **SIIMpaPrinterStatus** class.
According to the method in **SIIMpaPrinterStatus** class, the printer status can be retrieved.
For the method in **SIIMpaPrinterStatus** class, see "4.4.1(2) **SIIMpaPrinterStatus**".

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

startCallbackFunction — Start callback of printer status change

Starts the callback according to the printer status change.

Syntax

```
- (BOOL) startCallbackFunction: (SEL) method
                target: (NSObject*) target
                error: (NSError**) error;
```

Parameter

method	Selector specification Specify the selector of the execution method by the callback.
target	Object specification Specify the object including the method specified by **method**.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

The sample of execution method by the callback is follows.

```
- (BOOL) onStatusChanged: (SIIMpaPrinterStatus*) status
```

In **status**, the instance of **SIIMpaPrinterStatus** class includes the printer status at the point of time when it is changed is substituted. The printer status can be retrieved from **status** by the property of **SIIMpaPrinterStatus** class. See "4.4.1(2) **SIIMpaPrinterStatus**" for the property of **SIIMpaPrinterStatus** class.

stopCallbackFunction — Finish callback of printer status change

Finishes the callback started by **startCallbackFunction**.

Syntax

```
- (BOOL) stopCallbackFunciton: (NSError**) error;
```

Patameter

error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

registerStyleSheet — Register style sheet to printer

Registers the style sheet to the printer.

Syntax

```
- (BOOL) registerStyleSheet: (NSString*) filePath
            registerNumber: (NSInteger) regNum
            error: (NSError**) error;
```

Parameter

filePath	File path for style sheet
regNum	Style sheet number The range is 1 to 4.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description A maximum of 4 sheets can be registered.

The file extension for supporting style sheet is .css.

The maximum file size that can be registered is 1 MB (1048576 bytes).

The maximum number that can be registered in 1 style sheet is 64.

When specify a style sheet number which style sheet is already registered, the style sheet is overwritten.

For the available style sheet to register, see the printer command "Style Sheet Registration" in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

The timeout period specified in **setWriteTimeout** is valid.

deleteStyleSheet — Delete style sheet in printer

Deletes the style sheet registered to the printer.

Syntax

```
- (BOOL) deleteStyleSheet: (NSInteger) regNum
                    error: (NSError**) error;
```

Parameter

regNum		Style sheet number The range is 1 to 4.
error		Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

registerLogo — Register logo to printer

Registers the image data to a printer as a logo.
In syntax (a), dithering can be specified.
In syntax (b), dithering is fixed to be enabled.

Syntax

(a)-
```
(BOOL) registerLogo: (NSString*) filePath
              registerNumber: (NSInteger) regNum
              dithering: (Dithering) dithering
              error: (NSError**) error;
```

(b)-
```
(BOOL) registerLogo: (NSString*) filePath
              registerNumber: (NSInteger) regnum
              error: (NSError**) error;
```

Parameter

filePath		File path of image data
regNum		Logo number The range is 0 to 99.
dithering		Dithering See "4.4.2(3) **Dithering**" for available settings.
error		Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

The file extension for supporting image data is .bmp, .jpg, .jpeg, and .png.

The color image data is converted to monochrome image by binarization.

The maximum file size that can be registered is 1 MB (1048576 bytes).

The maximum image sizes that can be registered are 8192 dots in width and 2304 dots in height.

The registered situation for the logo registered by this method can be confirmed by executing **getPrinterInformation** or **getPrinterInformationString** with specifying **SII_MPM_GET_NV_MEM_KEYCODE_LIST** in **prnInfo**.

When specify a logo number which logo is already registered, the logo is overwritten.

The timeout period specified in **`setWriteTimeout`** is valid.

`deleteLogo` Delete logo in printer

Deletes the logo registered to the printer.

Syntax

```
- (BOOL) deleteLogo: (NSInteger) regNum
                error: (NSError**) error;
```

Parameter

`regNum`	Logo number The range is 0 to 99.
`error`	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **`NSError`** object specified by **`error.`**

YES	Normal
NO	Error

Description The logo deleted situation by this method can be confirmed by executing **`getPrinterInformation`** or **`getPrinterInformationString`** with specifying **SII_MPM_GET_NV_MEM_KEYCODE_LIST** in **`prnInfo`**.

`getPrinterInformation` `getPrinterInformationNumber` `getPrinterInformationString` Retrieve printer information

Retrieves the printer information.

Syntax

(a)
```
- (BOOL) getPrinterInformation: (PrinterInformation) prnInfo
          printerInformationValue: (NSData**) prnInfoValue
          error: (NSError**) error;
```

(b)
```
- (BOOL) getPrinterInformationNumber:
          (PrinterInformation) prnInfo
          printerInfoValue: (NSInteger**) prnInfoValue
          error: (NSError**) error;
```

(c)
```
- (BOOL) getPrinterInformationString:
          (PrinterInformation) prnInfo
          printerInformationValue: (NSString**) prnInfoValue
          error: (NSError**) error;
```

Parameter

`prnInfo`	Printer information type to retrieve See "4.4.2(4) **`PrinterInformation`**" for available settings and a list of printer information to retrieve. See "MP-A40 THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.
`prnInfoValue`	Printer information
`error`	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **`NSError`** object specified by **`error.`**

YES Normal
NO Error

Description Based on the specification in **prnInfo**, sends printer command to the printer for responding the printer information, analyzes the response data by the response extension, and stores it into **prnInfoValue** in the format of the numeric array, numerical value, and character string.

In all **PrinterInformation**, the printer information can be retrieved in the numeric array by using the syntax (a).

In the following **PrinterInformation**, the printer information can be retrieved in the numerical value by using the syntax (b).

- **SII_MPM_GET_NV_MEM_CAP**
- **SII_MPM_GET_NV_MEM_REM_CAP**
- **SII_MPM_GET_REM_USER_MEM_CAP**
- **SII_MPM_GET_REM_USER_MEM_CAP_DEFRAG**
- **SII_MPM_GET_PRN_ID_MODEL**
- **SII_MPM_GET_PRN_ID_TYPE**
- **SII_MPM_GET_PRN_ID_ROM_VER**
- **SII_MPM_GET_PRN_ID_FIRM_CHECKSUM_BOOT**
- **SII_MPM_GET_PRN_ID_FIRM_CHECKSUM_MAIN**
- **SII_MPM_GET_PRN_ID_FIRM_CHECKSUM**
- **SII_MPM_GET_MAINT_NUM_FEED_LINE**
- **SII_MPM_GET_MAINT_NUM_HEAD_ACTIVE**
- **SII_MPM_GET_MAINT_DRIVE_TIME**
- **SII_MPM_GET_MAINT_NUM_FEED_LINE_INTEGRATION**
- **SII_MPM_GET_MAINT_NUM_HEAD_ACTIVE_INTEGRATION**
- **SII_MPM_GET_MAINT_DRIVE_TIME_INTEGRATION**
- **SII_MPM_GET_HFONT_24_CHECKSUM**
- **SII_MPM_GET_HFONT_16_CHECKSUM**
- **SII_MPM_GET_FFONT_CHECKSUM**
- **SII_MPM_GET_FFONT_DATA_SIZE**
- **SII_MPM_GET_FFONT_ROM_ID**

In the following **PrinterInformation,** the printer information can be retrieved in the character string by using the syntax (c).

- **SII_MPM_GET_NV_MEM_KEYCODE_LIST**
- **SII_MPM_GET_PRN_ID_FIRM_VER_MAIN**
- **SII_MPM_GET_PRN_ID_MFR**
- **SII_MPM_GET_PRN_ID_MODEL_NAME**
- **SII_MPM_GET_PRN_ID_FIRM_VER_BOOT**
- **SII_MPM_GET_HFONT_24_ID**
- **SII_MPM_GET_HFONT_24_INT_CHAR**
- **SII_MPM_GET_HFONT_16_ID**
- **SII_MPM_GET_HFONT_16_INT_CHAR**
- **SII_MPM_GET_FFONT_LANG**
- **SII_MPM_GET_FFONT_STANDARD**
- **SII_MPM_GET_FFONT_COMPANY**

The timeout period specified in **setResponseTimeout** is valid.

This method is aborted by **reset**.

getVersion — Retrieve SDK version

Retrieves the SDK version as a character string.

Syntax
```
- (NSString*) getVersion;
```

Return value SDK version character string (Example: When the SDK version is Ver.1.0.0, the return value is "1.0.0")

Description This method can be executed regardless of whether **isOpened** is YES or NO.

controlTransaction — Start/finish print data batch sending

Buffers the print data in the target method for batch process and sends the data in buffer to the printer.

Syntax
```
- (BOOL) controlTransaction: (TransactionFunction) control
                      error: (NSError**) error;
```

Parameter

control	Operation selection of batch process See "4.4.2(5) **TransactionFunction**" for available settings.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description The target method for the batch process is as follows.

- **write**
- **selectStandardMode**
- **selectPageMode**
- **printPageModeData**
- **setPageModeVerticalPosition**
- **setStandardModeArea**
- **setStandardModeAlignment**
- **setHorizontalPosition**
- **setStandardModeBarcodeDirection**
- **setLineSpacing**
- **setCharacterRightSpace**
- **setCharacterFormatting**
- **printText**
- **printLogo**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printBarcode**
- **print2DCode**
- **printPageModeRectangle**
- **feedLine**
- **feedDotLine**
- **feedCutPosition**
- **feedMarkPosition**

By **control**, buffering of the print data in the target method for batch process is started or finished.

・Start buffering
When this method with **SII_MPM_TRANSACTION_START** in **control** is executed, the buffering of the print data in the target method for batch process is started. The print data in the target method for batch process executed during buffering is not sent to the printer but stored in the buffer. Any method other than the target method for batch process is immediately executed.

・Finish buffering
When this method with **SII_MPM_TRANSACTION_PRINT** in **control** is executed, the buffering of the print data in the target method for batch process is finished, and the data in the buffer is sent to the printer.
When this method with **SII_MPM_TRANSACTION_CLEAR** in **control** is executed, the buffering is interrupted and the data in the buffer is discarded.

When this method with **SII_MPM_TRANSACTION_PRINT** or **TRANSACTION_CLEAR** in **control** is executed without starting buffering, an error occurs.

When this method with **SII_MPM_TRANSACTION_START** in **control** is executed during buffering, the data in the buffer is discarded but the buffering is continued.

When no data exists in the buffer and this method with **SII_MPM_TRANSACTION_PRINT** in **control** is executed, the data is not sent to the printer and the buffering is finished.

When this method with **SII_MPM_TRANSACTION_PRINT** in **control** is being executed and this method with **SII_MPM_TRANSACTION_START** in **control** is executed from another thread, the data sending by this method with **SII_MPM_TRANSACTION_PRINT** in **control** is continued, this method with **SII_MPM_TRANSACTION_START** in **control** is executed from another thread, and the buffering is newly started.

The maximum buffer size depends on the system.
When the data in the buffer exceeds the maximum size, the execution of the target method for batch process becomes error, but the data in the buffer is stored.

When the buffered data is sending by **SII_MPM_TRANSACTION_PRINT**, the timeout period specified in **setWriteTimeout** is valid.

When an error occurs during sending data by **SII_MPM_TRANSACTION_PRINT**, the remained data for sending is discarded.

When **reset** is executed during sending data **SII_MPM_TRANSACTION_PRINT**, the remained data for sending is discarded.

selectStandardMode — Select standard mode

Starts standard mode.

Syntax
```
- (BOOL) selectStandardMode: (NSError**) error;
```

Parameter

error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description Right after execution of **open**, standard mode is selected.

When page mode is interrupted by **selectPageMode**, the page mode data is discarded.

selectPageMode — Start page mode

Starts page mode and specify the print area and the direction.

Syntax

```
- (BOOL) selectPageMode: (NSInteger) startX
                 startY: (NSInteger) startY
                 width: (NSInteger) width
                 height: (NSInteger) height
                 direction: (Direction) direction
                 error: (NSError**) error;
```

Parameter

startX	Horizontal start point (dot) The range is 0 to 2399.
startY	Vertical start point (dot) The range is 0 to 2399.
width	Print area width (dot) The range is 1 to 2400.
width	Print area height (dot) The range is 1 to 2400.
direction	Print direction (dot) See "4.4.2(6) **Direction**" for available settings.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

When **startX** exceeds the print width, the print area setting by this method is disabled.
When (**startX** + **width**) exceeds the print width, **startX** is set to the specified value and **width** is set to (print width – **startX**).
When (**startY** + **height**) is 2400 or more, **startY** is set to the specified value and **height** is set to (2400 – **startY**).
The print width can be specified in the function setting. For the print width and the function setting, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

Layout the character, image, barcode, and 2-dimensional barcode in the optional print position within the specified print area and configure the page mode data by **setHorizontalPosition** and **setPageModeVerticalPosition**.

When page mode is interrupted by **selectStandardMode**, the page mode data stored in the printer will be discarded.

When the printer is reset, the setting is back to standard mode.

printPageModeData — Print page mode data

Prints the page mode data.

Syntax

```
- (BOOL) printPageModeData: (NSError**) error;
```

Parameter

error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description Execute this method after starting page mode by **selectPageMode** and configuring the page mode data within the print area.

This method is ignored in standard mode.

setPageModeVerticalPosition Specify vertical absolute position in page mode

Specify the absolute position of print data in vertical direction within the print area in page mode.

Syntax
```
- (BOOL) setPageModeVerticalPosition:
              (NSInteger) verticalPosition
               error: (NSError**) error;
```

Parameter

verticalPosition	Data mapping starting position in vertical direction (dot) The range is 0 to 2399.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description The start point varies depending on the print direction in page mode specified for **direction** in **selectPageMode**. For the print direction and the start point, see "4.4.2(6) **Direction**".

The setting of this method is disabled in standard mode.

When the specified value exceeds the print area, it is ignored.

When the printer is reset, the setting in this method is disabled.

setStandardModeArea Specify print area in standard mode

Specify the print area in standard mode.

Syntax
```
- (BOOL) setStandardModeArea: (NSInteger) leftMargin
                printAreaWidth: (NSInteger) prnAreaWidth
                error: (NSError**) error;
```

Parameter

leftMargin	Left margin (dot) The range 0 to 831.
prnAreaWidth	Print area width (dot) The range 1 to 832.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description The relation between left margin and print area width is shown in Figure 4-1. The print data is mapped to the shaded print area.

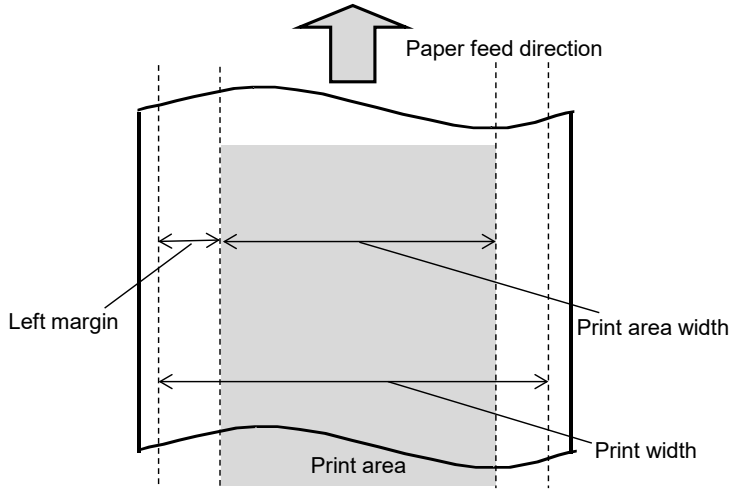


Figure 4-1

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the specified value in **`leftMargin`** exceeds the print width, the value is set to the print width. When the specified value in **`prnAreaWidth`** exceeds the print width, the value is set to the difference between the print width and the left margin.
The print width can be specified in the function setting. For the print width and the function setting, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

When the printer is reset, **`leftMargin`** is set to 0 and **`prnAreaWidth`** is intialized to the print width.

`setStandardModeAlignment` Alignment

Set the alignment in standard mode.

Syntax
```
- (BOOL) setStandardModeAlignment: (Alignment) align
                    error: (NSError**) error;
```

Parameter

`align`		Alignment See "4.4.2(7) **`Alignment`**" for available settings.
`error`		Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **`NSError`** object specified by **`error.`**

YES	Normal
NO	Error

Description When the print area is specified in **`setStandardModeArea`**, the position is aligned within the specified print area.

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the printer is reset, the setting in this method is back to the default value.

setHorizontalPosition — Specify absolute position

Specify the absolute horizontal position.

Syntax

```
- (BOOL) setHorizontalPosition: (NSInteger) horizontalPosition
                  error: (NSError**) error;
```

Parameter

horizontalPosition	Horizontal print starting position (dot) The range is 0 to 2399.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

In standard mode, the left margin is based on the one specified in **setStandardModeArea**.

In page mode, the start point changes depend on the specified print direction in **direction** of **selectPageMode**. For print direction and start point, see "4.4.2(6) **Direction**".

The value specified in **setStandardModeArea** and **selectPageMode** that exceeds the print area is ignored.

When the printer is reset, the setting in this method is disabled.

setStandardModeBarcodeDirection — Select print direction for barcode or 2-dimensional barcode

Specify the print direction for barcode or 2-dimensional barcode in standard mode.

Syntax

```
- (BOOL) setStandardModeBarcodeDirection: (Rotate) rotate
                  error: (NSError**) error;
```

Parameter

rotate	Print direction See "4.4.2(15) **Rotate**" for available settings.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

By executing **printBarcode** or **print2DCode**, the setting of this method is back to the default value.

The setting of this method is disabled in page mode. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.

When **SII_MPM_ROTATE_90_TO_RIGHT** or **SII_MPM_ROTATE_90_TO_LEFT** is specified in **rotate**, the available print width of barcode or 2-dimensional barcode is max. 300 mm. When the value exceeds 300 mm, barcode or 2-dimensional barcode is not printed.

When the printer is reset, the setting in this method is back to the default value.

setLineSpacing — Specify line space amount

Specify the line spacing.

Syntax

```
- (BOOL) setLineSpacing: (NSInteger) lineSpacing
                  error: (NSError**) error;
```

Parameter

lineSpacing	Line spacing (dot) The range is 0 to 255.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

The line spacing can be independently set in the standard mode and the page mode.

When the printer is reset, **lineSpacing** is set to 34.

setCharacterRightSpace — Specify character right space amount

Specify the character right space amount.

Syntax

```
- (BOOL) setCharacterRightSpace: (NSInteger) space
                           error: (NSError**) error;
```

Parameter

space	Right space amount (dot) The range is 0 to 255.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

The right space amount can be independently set in the standard mode and the page mode.

This method is enabled for 1-byte characters and 2-byte characters. For 2-byte characters, the left space is set at 0.

When the printer is reset by **reset** or turned off, **space** is set to 0.

selectCharacterSet — Select character set

Select the character set.

Syntax
```
- (BOOL) selectCharacterSet: (CharacterSet) charSet
                      error: (NSError**) error;
```

Parameter

charSet		Character set See "4.4.2(8) **CharacterSet**" for available settings.

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description According to the language setting in iOS device, the default setting of the character set is as follows.

Japanese:	**SII_MPM_CODEPAGE_KATAKANA**
Other than Japanese:	**SII_MPM_CODEPAGE_1252**

When this method is executed and then the printer is reset by **reset** or turned off, the setting in this method is enabled.

getCharacterSet — Retrieve specified character set

Retrieves the specified character set.

Syntax
```
- (NSInteger) getCharacterSet;
```

Return value Character set

Description For the available setting to retrieve, see "4.4.2(8) **CharacterSet**".

selectInternationalCharacterSet — Select international character set

Select the international character set.

Syntax
```
- (BOOL) selectInternationalCharacterSet:
                    (InternationalCharacterSet) intCharSet
                     error: (NSError**) error;
```

Parameter

intCharSet		International character set See "4.4.2(9) **InternationalCharacterSet**" for available settings.

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description According to the language setting in iOS device, the default setting of the International character set is as follows.

Japanese:	**SII_MPM_INT_CHAR_SET_JAPAN**
Other than Japanese:	**SII_MPM_INT_CHAR_SET_USA**

When this method is executed and then the printer is reset by **reset** or turned off, the setting in this method is enabled.

getInternationalCharacter — Retrieve specified international character set

Retrieves the specified international character set.

Syntax
```
- (NSInteger) getInternationalCharacterSet;
```

Return value International character set

Description For the available setting to retrieve, see "4.4.2(9) **InternationalCharacterSet**".

setCharacterFormatting — Format character

Set the character formatting.

Syntax
```
- (BOOL) setCharacterFormatting: (CharacterType) type
                  verticalScale: (CharacterScale) verticalScale
                horizontalScale: (CharacterScale) horizontalScale
                      underline: (Underline) underline
                           bold: (Bold) bold
                        reverse: (Reverse) reverse
                         rotate: (Rotate) rotate
                          error: (NSError**) error;
```

Parameter

type Character font
See "4.4.2(10) CharacterType" for available settings.

verticalScale Character size (in vertical direction)
See "4.4.2(11) CharacterScale" for available settings.

horizontalScale Character size (in horizontal direction)
See "4.4.2(11) CharacterScale" for available settings.

underline Underline
See "4.4.2(12) Underline" for available settings.

bold Bold print
See "4.4.2(13) Bold" for available settings.

reverse Reverse print
See "4.4.2(14) **Reverse**" for available settings.

rotate Character Rotation Print
In page mode, the setting in *rotate* is disabled. When this method is executed in page mode, the setting is reflected to the print after starting standard mode.
In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.
In standard mode, when **SII_MPM_ROTATE_90_TO_RIGHT** or **SII_MPM_ROTATE_90_TO_LEFT** is specified, the operation is as follows:
- Underline print is not performed.
- the scale of height and width in **verticalScale** and **horizontalScale** are reversed.

See "4.4.2(15) **Rotate**" for available settings.

error Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description When the printer is reset, the setting in this method back to the default value.

printText Send text data

Sends the text data.

Syntax
```
- (BOOL) printText: (NSString*) text
                 error: (NSError**) error;
```

Parameter **text** Text data to send to the printer

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description This method sends the text data specified in **text** after encoding it to recognizable text data for the printer based on the character set specified in **selectCharacterSet**.

Based on the setting in **selectCharacterSet** and **selectInternationalCharacterSet**, printer commands "Character Code Table Select", "International Character Select", "Kanji Code System Selection", and "Kanji Mode Cancel" are added to the send data as the header.
For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method does not add any line feed code in the last of the text data.

The maximum data size that can be specified is 16 KB (16384 bytes).

When **controlTransaction** is not used, the timeout period specified in **setWriteTimeout** is valid.

This method is aborted by **reset**.

printLogo Print logo registered in printer

Prints the logo registered in the printer.

Syntax
```
- (BOOL) printLogo: (NSInteger) regNum
                 error: (NSError**) error;
```

Parameter **regNum** Logo number
The range is 0 to 99.

error Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description Register the logo by **registerLogo**.

When the specified logo is not registered in **regNum**, this method is ignored.

setImageTransactionMode — Specify image printing mode

Sets the print mode of image.

Syntax
```
- (void) setImageTransactionMode: (ImageTransactionMode) mode;
```

Parameter

mode	Print mode of image See "4.4.2(26) **ImageTransactionMode**" for available settings.
error	Error object reference

Description The default setting of the print mode of image is **SII_MPM_IMAGE_TRANSACTION_MODE_OFF**.

When this method is executed and then the printer is reset by **reset** or turned off, the setting in this method is enabled.

This method is enabled in the following methods.
- **sendDataFile**
- **sendImage**
- **printPDF**

getImageTransactionMode — Retrieve image printing mode

Retrieves the currently set print mode of image.

Syntax
```
- (BOOL) getImageTransactionMode;
```

Return value Print mode of image

YES **SII_MPM_IMAGE_TRANSACTION_MODE_ON**
NO **SII_MPM_IMAGE_TRANSACTION_MODE_OFF**

Description For the available setting to retrieve, see "4.4.2(26) **ImageTransactionMode**".

sendDataFile — Send file data

Sends the file data to the printer.
In syntax (a), dithering can be specified.
In syntax (b), dithering is fixed to enabled.

Syntax

(a)
```
- (BOOL) sendDataFile: (NSString*) filePath
            dithering: (Dithering) dithering
            error: (NSError**) error;
```

(b)
```
- (BOOL) sendDataFile: (NSString*) filePath
            error: (NSError**) error;
```

Parameter

filePath	File path to send to the printer

dithering	Dithering It is enabled when the extension of the file specified at `filePath` is .bmp, .jpg, .jpeg, or .png. See "4.4.2(3) **Dithering**" for available settings.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description Depending on the specified file extension, the followings are processed.

・When the file extension is .bmp, .jpg, .jpeg, or .png:
The image data is converted to printable format for the printer and is sent to the printer.
The color image data is converted to monochrome image by binarization.
If **setImageTransactionMode** is set to **SII_MPM_IMAGE_TRANSACTION_MODE_ON**, printing is performed in batch after mapping the image file on the memory of the printer.

・When the file extension is .txt:
Text data format supports UTF-8.
This method sends the text data after encoding it to recognizable text data for the printer based on the character set specified in **selectCharacterSet**.
Based on the setting in **selectCharacterSet** and **selectInternationalCharacterSet**, printer commands "Character Code Table Select", "International Character Select", "Kanji Code System Selection", and "Kanji Mode Cancel" are added to the send data as the header.
For the details of printer commands, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".
This method does not add any line feed code in the last of the text data.

・When the file extension is .bin, or.dat
The text data is sent to printer without conversion.

・When the file extension is htm, or .html
The text data is sent to printer without conversion.
Be sure to add an HTML end tag in the last of file data.

The maximum file size that can be specified is 16 MB (16777216 bytes).

When **controlTransaction** is not used, the timeout period specified in **setWriteTimeout** is valid.

Do not include a printer command that initializes the printer other than the printer command "Initialize Printer" in the data to send. When perform Hardware Reset, use reset. For printer initialization, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

This method is aborted by **reset**.

sendImage — Send image

Sends image data to the printer.

Syntax
```
- (BOOL) sendImage: (UIImage*) image
                dithering: (Dithering) dithering
                error: (NSError**) error;
```

Parameter

image	UIImage-type object to be sent to the printer. Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and registered. If **setImageTransactionMode** is set to **SII_MPM_IMAGE_TRANSACTION_MODE_ON**, printing is performed in batch after mapping the image file on the memory of the printer.
dithering	Dithering See "4.4.2(3) **Dithering**" for available settings.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description When **controlTransaction** is not used, the timeout period specified in **setWriteTimeout** is valid.

This method is aborted by **reset**.

printPDF — Print PDF page

Prints the specified page of the selected PDF file

Syntax

```
- (BOOL) printPDF: (NSString*) filePath
                startIndex: (NSInteger) startIndex
                endIndex: (NSInteger) endIndex
                rotate: (Rotate) rotate
                width: (NSInteger) width
                bottomMargin: (NSInteger) bottomMargin
                dithering: (Dithering) dithering
                error: (NSError**) error;
```

Parameter

filePath	File path of PDF
startIndex	Start number of the printed page The range is -1, 1 to the number of pages in the PDF file.
endIndex	End number of the printed page The range is 1 to 2147483647.
rotate	Rotation direction of the image See "4.4.2(15) **Rotate**" for available settings.
width	Width of the image (dot) The range is -1, 1 to 832.
bottomMargin	Bottom margin (dot) The range is -1, 0 to 2400.
dithering	Dithering See "4.4.2(3) **Dithering**" for available settings.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

PDF is converted to printable format for the printer and is sent to the printer.

The file extension for supporting PDF is .pdf.
The maximum file size that can be specified is 16 MB (16777216 bytes).
The color image data is converted to monochrome image by binarization.
If **setImageTransactionMode** is set to **SII_MPM_IMAGE_TRANSACTION_MODE_ON**, printing is performed in batch after mapping the image file on the memory of the printer.

When -1 is specified for **startIndex**, all pages are printed.
When -1 is specified for **startIndex**, the value of **endIndex** is ignored.
When 1 or more is specified for **startIndex**, pages from the page number specified in **startIndex** to the page number specified in **endIndex** are printed.
Up to 100 pages of PDF can be printed at a time regardless of the page number to be specified.
When a value more than the value specified for **endIndex** is specified for **startIndex**, an error occurs.
When a value more than the number of pages in the PDF file is specified for **endIndex**, pages from the page number specified in **startIndex** to the last page of the PDF file are printed.

When an out-of-range value is specified for **startIndex** or **endIndex**, an error occurs.
When -1 is specified for **width**, size conversion is not performed.
When a value between 1 and 832 is specified for **width**, the image width is converted to the specified value while maintaining the aspect ratio.
When -1 is specified for **bottomMargin**, the image is created and printed while maintaining the bottom margin.
When a value between 0 and 2400 is specified for **bottomMargin**, the bottom margin is changed to the specified size.
When a value between 0 and 2400 is specified for **bottomMargin**, blank pages are not printed.
When marked paper is selected, marked paper form feed is performed after printing the PDF.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about how to use the marked paper.

When **controlTransaction** is not used, the send timeout period for this method is the value set by **setWriteTimeout** for each page printed.

This method is aborted by **reset**.

Note

・This method is not supported in page mode.
・No guarantee of printing operation when more than 100 pages are printed at a time.

Prints the barcode.

Syntax

(a) – `(BOOL)` **`printBarcode:`** `(TypeBarcode) type`
`text: (NSString*) text`
`moduleWidth: (ModuleWidthBarcode) moduleWidth`
`moduleHeight: (NSInteger) moduleHeight`
`hri: (HriPositionBarcode) hri`
`nwRatio: (NwRatioBarcode) nwRatio`
`error: (NSError**) error;`

(b) – `(BOOL)` **`printBarcode:`** `(TypeBarcode) type`
`binary: (NSData*) binary`
`moduleWidth: (ModuleWidthBarcode) moduleWidth`
`moduleHeight: (NSInteger) moduleHeight`
`hri: (HriPositionBarcode) hri`
`nwRatio: (NwRatioBarcode) nwRatio`
`error: (NSError**) error;`

Parameter

`type` Barcode type
See "4.4.2(16) **`TypeBarcode`**" for available settings.
(a) **`text`** : Barcode text data
(b) **`binary`** : Barcode binary data

The available values in **`text`** and **`binary`** are as follows.

Barcode Type	Data Size	Barcode Data to Input	
UPC-A	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
UPC-E	11 to 12 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13)	12 to 13 bytes	'0' to '9'	(0x30 to 0x39)
JAN8 (EAN8)	7 to 8 bytes	'0' to '9'	(0x30 to 0x39)
CODE39	1 to 150 bytes	' '	(0x20)
		'$'	(0x24)
		'%'	(0x25)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' to '9'	(0x30 to 0x39)
		'A' to 'Z'	(0x41 to 0x54)
ITF	2 to 15 bytes (Even number only)	'0' to '9'	(0x30 to 0x39)

Barcode Type	Data Size	Barcode Data to Input	
CODABAR	1 to 150 bytes	'$'	(0x24)
		'+'	(0x2B)
		'-'	(0x2D)
		'.'	(0x2E)
		'/'	(0x2F)
		'0' to '9'	(0x30 to 0x39)
		':'	(0x3A)
		'A' to 'D'	(0x41 to 0x44)
CODE93	1 to 150 bytes		(0x00 to 0x7F)
CODE128	2 to 150 bytes		(0x00 to 0x7F)*1
JAN13 (EAN13) add-on 2	14 to 15 bytes	'0' to '9'	(0x30 to 0x39)
JAN13 (EAN13) add-on 5	17 to 18 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Omni-directional	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Truncated	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Limited	13 bytes	'0' to '9'	(0x30 to 0x39)
GS1 Databar Expanded	2 to 150 bytes	' 'to '"'	(0x20 to 0x22)
		'%' to '?'	(0x25 to 0x3F)
		'A' to 'Z'	(0x41 to 0x5A)
		'_'	(0x5F)
		'a' to 'z'	(0x61 to 0x7A)
		'{'	(0x7B)

*1: The available barcode for input varies depending on the specified code set.
CODE A : 0x00 to 0x50
CODE B : 0x00 to 0x7F
CODE C : 2 digit numbers from 00 to 99 (0x00 to 0x63)

`moduleWidth` Barcode module width or fine element
See "4.4.2(17) **`ModuleWidthBarcode`**".

`moduleHeight` Barcode module height(dot)
See the following table for available settings.

Barcode Type	`moduleHeight` **(dot)**
UPC-A	0: Initial value (162 dots), 1 to 255
UPC-E	
JAN13 (EAN13)	
JAN8 (EAN8)	
CODE39	
ITF	
CODABAR	
CODE93	
CODE128	
JAN13 (EAN13) add-on 2	
JAN13 (EAN13) add-on 5	

Barcode Type	moduleHeight (dot)	
GS1 Databar Omni-directional	0: Initial value (162 dots)	66 to 255*1
		99 to 255*2
		132 to 255*3
		165 to 255*4
		198 to 255*5
GS1 Databar Truncated	0: Initial value (162 dots)	26 to 255*1
		39 to 255*2
		52 to 255*3
		65 to 255*4
		78 to 255*5
GS1 Databar Limited	0: Initial value (162 dots)	20 to 255*1
		30 to 255*2
		40 to 255*3
		50 to 255*4
		60 to 255*5
GS1 Databar Expanded	0: Initial value (162 dots)	68 to 255*1
		102 to 255*2
		136 to 255*3
		170 to 255*4
		204 to 255*5

*1: When **moduleWidth** is **SII_MPM_MODULE_WIDTH_BARCODE_2**.
*2: When **moduleWidth** is **SII_MPM_MODULE_WIDTH_BARCODE_3**.
*3: When **moduleWidth** is **SII_MPM_MODULE_WIDTH_BARCODE_4**.
*4: When **moduleWidth** is **SII_MPM_MODULE_WIDTH_BARCODE_5**.
*5: When **moduleWidth** is **SII_MPM_MODULE_WIDTH_BARCODE_6**.

hri Barcode HRI character
See "4.4.2(18) HriPositionBarcode" for available settings.

nwRatio Barcode N:W ratio
The available setting varies depending on the setting of **type**.
For the details, see "4.4.2(19) **NwRatioBarcode**".

error Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

When the barcode width exceeds the print area, the barcode is not printed.

This method sends the text data specified in syntax (a) **text** after encoding it to recognizable text data for the printer.
The binary data specified in syntax (b) **binary** is sent to the printer as it is.

Special Codes specified in CODE128 are as follows.

Input Data		Special Code per Code Set		
		CODE A	CODE B	CODE C
'{S'	(0x7B, 0x53)	SHIFT	SHIFT	SHIFT
'{A'	(0x7B, 0x41)	-	CODE A	CODE A
'{B'	(0x7B, 0x42)	CODE B	-	CODE B
'{C'	(0x7B, 0x43)	CODE C	CODE C	-
'{1'	(0x7B, 0x31)	FNC1	FNC1	FNC1
'{2'	(0x7B, 0x32)	FNC2	FNC2	FNC2
'{3'	(0x7B, 0x33)	FNC3	FNC3	FNC3
'{4'	(0x7B, 0x34)	FNC4	FNC4	FNC4
'{{'	(0x7B, 0x7B)	-	'{'	-

Check digits for the following barcode are automatically calculated.
・UPC-A
・UPC-E
・JAN13 (EAN13)
・JAN8 (EAN8)
・JAN13 (EAN13) add-on 2
・JAN13 (EAN13) add-on 5
・GS1 Databar Omni-directional
・GS1 Databar Truncated
・GS1 Databar Limited

When the data including the check digit is specified for the following barcodes, the check digit is ignored and recalculated on the printer.
・UPC-A
・UPC-E
・JAN13 (EAN13)
・JAN8 (EAN8)

Specifying the start code and stop code ('*') of CODE39 is not required since the codes are added automatically.

When odd number of data is specified in ITF barcode data, the last data is discarded.

In CODABAR barcode data, input 'A' to 'D' as the start code and stop code.

In GS1 Databar Expanded barcode data, use '{' only for specifying FNC1. When specify FNC1, input '{1'(0x7B, 0x31).

When specified value in any one of **`text`**, **`binary`**, **`moduleWidth`**, **`moduleHeight`**, or **`nwRatio`** is not corresponded to the barcode type specified in **`type`**, an error occurs.

When **`controlTransaction`** is not used, the timeout period specified in **`setWriteTimeout`** is valid.

print2DCode — Print 2-dimensional barcode

Prints the 2-dimensional barcode.

Syntax

(a) –
```
(BOOL) print2DCode: (Type2DCode) type
                    text: (NSString*) text
                    mode: (Mode2Dcode) mode
                    moduleSize: (ModuleSize2Dcode) moduleSize
                    moduleHeight: (NSInteger) moduleHeight
                    column: (NSInteger) column
                    row: (NSInteger) row
                    errorCorrect:(ErrorCorrect2Dcode) errorCorrect
                    error: (NSError**) error;
```

(b) –
```
(BOOL) print2DCode: (Type2DCode) type
                    binary: (NSData*) binary
                    mode: (Mode2Dcode) mode
                    moduleSize: (ModuleSize2Dcode) moduleSize
                    moduleHeight: (NSInteger) moduleHeight
                    column: (NSInteger) column
                    row: (NSInteger) row
                    errorCorrect:(ErrorCorrect2Dcode) errorCorrect
                    error: (NSError**) error;
```

Parameter

type — 2-dimensional barcode type
See "4.4.2(20) Type2DCode" for available settings.
(a) **text** : Barcode text data
(b) **binary** : Barcode binary data

mode — 2-dimensional barcode mode
The available setting varies depending on the setting of **type**.
For the details, see "4.4.2(21) Mode2Dcode".
When select **SII_MPM_MODE_2DCODE_MAXI_CODE_2**, add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) data in the beginning of **text** or **binary**.
When select **SII_MPM_MODE_2DCODE_MAXI_CODE_3**, add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) data in the beginning of **text** or **binary**.

moduleSize — 2-dimensional barcode module size
The available setting varies depending on the setting of **type**.
For the details, see "4.4.2(22) **ModuleSize2DCode**".

moduleHeight — 2-dimensional barcode module height (dot)
moduleHeight is enabled in PDF417 and GS1 Databar Stacked Omni-directional.
・PDF417
Specifies the row height of PDF. The valid range is 2 to 127.
・GS1 Databar Stacked Omni-directional
Specifies the height of 1 row. The valid range is 33 to 255.

	column	Number of columns in 2-dimensional barcode **column** is enabled in PDF417 and GS1 Databar Expanded Stacked. ・PDF417 Specifies the number of columns in the data area. The valid range is 0 to 30. When 0 is specified, the number of column is set automatically. ・GS1 Databar Stacked Expanded Specifies an even number of columns per line. The valid range is 2 to 20.
	row	Number of rows in 2-dimensional barcode **row** is enabled in PDF417 only. The valid range is 0, and 3 to 90. When 0 is specified, the number of row is set automatically.
	errorCorrect	Error correction level of 2-dimensional barcode The available setting varies depending on the setting of **type**. For the details, see "4.4.2(23) **ErrorCorrect2DCode**".
	error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

This method sends the text data specified in syntax (a) **text** after encoding it to recognizable text data for the printer.
In **text**, input the corresponded character data in ASCII style, according to the standard 2-dimensional barcode type specified in **type**.
The binary data specified in syntax (b) **binary** is sent to the printer as it is.
In **binary**, input the corresponded character data in ASCII style character code, according to the standard 2-dimensional barcode type specified in **type**.

When specified value in any one of **text**, **binary**, **mode**, **moduleSize**, **moduleHeight**, **column**, or **errorCorrect** is not corresponded to the 2-dimensional barcode type specified in **type**, an error occurs.

When the 2-dimensional barcode width exceeds the print area width, this method is ignored.

When **controlTransaction** is not used, the timeout period specified in **setWriteTimeout** is valid.

printPageModeRectangle — Draw rectangular

Draws a rectangle in page mode.

Syntax

```
- (BOOL) printPageModeRectangle: (NSInteger) startX
                    startY: (NSInteger) startY
                    width: (NSInteger) width
                    height: (NSInteger) height
                    thickness: (NSInteger) thickness
                    error: (NSError**) error;
```

Parameter

startX	Horizontal start point (dot) The range is 0 to 2395.
startY	Vertical start point (dot) The range is 0 to 2395.
width	Width (dot) The range is 4 to 2400.
height	Height (dot) The range is 4 to 2400.
thickness	Line width (dot) The range is 2 to 40.
error	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description

Do not execute this method in standard mode.

The start point varies depending on the print direction in page mode specified in **direction** of **selectPageMode**.
For the details of print direction and start point, see "4.4.2(6) **Direction**".

When the setting in **startX**, **startY**, **width**, or **height** exceeds the print area in page mode specified in **selectPageMode**, the exceeded area in print area is not drawn.

In this method, the printer command "Ruled Line OFF" and "Ruled Line Buffer Clear" are sent, and the ruled line is changed to OFF and ruled line buffer is cleared. For the details of printer command, see "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

feedLine — Feed paper by line

Feeds the paper by line.

Syntax

```
- (BOOL) feedLine: (NSInteger) lines
                    error: (NSError**) error;
```

Parameter

lines	Number of lines to feed (line) The range is 0 to 255.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description In standard mode, feeds the paper by the specified number of lines.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.
In page mode, the vertical data mapping starting position is shifted by the specified number of lines.

The paper feed length for 1 line (line feed space) can be specified in **setLineSpacing**.

feedDotLine — Feed paper by dot

Feed the paper by dot.

Syntax
```
- (BOOL) feedDotLine: (NSInteger) dotLines
                error: (NSError**) error;
```

Parameter

dotLines	Number of dots to feed (dot) In the standard mode, the range is -48 to 8192. In page mode, the range is 0 to 8192. When the specified value is within -48 to -1, this method is ignored.
error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description In standard mode, feeds the paper by the specified number of dots.
When data exists in the line buffer of the printer, paper feed is performed after printing 1 line.
In page mode, the vertical data mapping starting point is shifted by the specified number of dots.

feedCutPosition — Feed paper to cut position

Feeds the paper to the paper cut position.

Syntax
```
- (BOOL) feedCutPosition: (NSError**) error;
```

Parameter

error	Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description In standard mode, when the print data remains in the printer, this method is not executed.
Execute this method after the printing is completed.

feedMarkPosition — Marked paper form feed

Performs the marked paper form feed and the form feed position correct.

Syntax
```
- (BOOL) feedMarkPosition: (NSInteger) dotLines
                    error: (NSError**) error;
```

Parameter

dotLines	Correction amount (dot) The range is -48 to 255. When correction is not needed, specify 0.
error	Error object reference

Return value — The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description — In standard mode, when the print data remains in the printer, this method is not executed. Execute this method after the printing is completed.

This method is effective only when marked paper is selected.
See "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about how to use the marked paper.

setLog — Specify log output

Specify the log output.

Syntax
```
- (BOOL) setLog: (NSInteger) logLevel
                 logFileSize: (LogFileSize) logFileSize
                 error: (NSError**) error;
```

Parameter

logLevel	Log output level Specify 0. When other than 0 is specified, the contents of output log is not guaranteed.
logFileSize	Maximum size of log file See "4.4.2(24) **LogFileSize**" for the available setting.
error	Error object reference

Return value — The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES	Normal
NO	Error

Description — Log file is saved as a local file of iOS application with programmed this library.

Log file name: PrinterManager.log.x (x range is 0 to 4)

First of all, the log file is created called PrinterManager.log.0.
When the file size exceeds the maximum size of the log file, changes the file name to PrinterManager.log.1, and creates the new file called PrinterManager.log.0.
The number of created file is up to 5.

- **Public Property**

isOpened — Retrieve printer using status

Retrieves that the printer's open status by **open**.

Syntax `@property(readonly) BOOL` **`isOpened;`**

Parameter **`isOpened`** Using printer

Return value
YES Using printer is started
NO Using printer is not started.

(2) **SIIMpaPrinterStatus**

- **Summary**

This class stores the printer status.
This class provides following functions.

Public Properties

Method	Function Summary
errOffline	Retrieve offline error status
errHardware	Retrieve hardware error status
errVoltage	Retrieve Vp Voltage error status
errHeadTemperature	Retrieve head temperature error status
errOutOfPaper	Retrieve out-of-paper error status
errMarkPaperJam	Retrieve paper jam error status while detecting marked paper.
errCoverOpen	Retrieve paper cover open error status
errBattery	Retrieve battery error status
stateFeedSwitch	Retrieve feed switch status
statePaperFeed	Retrieve paper feed status
stateReturnWaiting	Retrieve return waiting status
stateFlashMemoryRewriting	Retrieve FLASH memory rewrite status
stateBattery	Retrieve battery voltage status

- **Public Properties**

errOffline — Retrieve offline error status

Retrieves the offline error status.

Syntax `@property(readonly) BOOL` **`errOffline;`**

Return value
YES Offline error
NO Online

errHardware — Retrieve hardware error status

Retrieves the hardware error status.

Syntax `@property(readonly) BOOL` **`errHardware;`**

Return value
YES error
NO OK

errVoltage — Retrieve Vp Voltage error status

Retrieves the Vp Voltage error status.

Syntax `@property(readonly) BOOL **errVoltage;**`

Return value
YES error
NO OK

errHeadTemperature — Retrieve head temperature error status

Retrieves the head temperature error status.

Syntax `@property(readonly) BOOL **errHeadTemperature;**`

Return value
YES error
NO OK

errOutOfPaper — Retrieve out-of-paper error status

Retrieves the out-of-paper error status.

Syntax `@property(readonly) BOOL **errOutOfPaper;**`

Return value
YES error
NO OK

errMarkPaperJam — Retrieve paper jam error status while detecting marked paper.

Retrieves the paper jam error status while detecting marked paper.

Syntax `@property(readonly) BOOL **errMarkPaperJam;**`

Return value
YES error
NO OK

errCoverOpen — Retrieve paper cover open error status

Retrieves the paper cover open error status.

Syntax `@property(readonly) BOOL **errCoverOpen;**`

Return value
YES error
NO OK

errBattery — Retrieve battery error status

Retrieves the battery error status.

Syntax `@property(readonly) BOOL **errBattery;**`

Return value
YES error
NO OK

stateFeedSwitch — Retrieve feed switch status

Retrieves the feed switch status.

Syntax `@property(readonly) BOOL **stateFeedSwitch;**`

Return value

YES	ON
NO	OFF

statePaperFeed — Retrieve paper feed status

Retrieves the paper feed status.

Syntax `@property(readonly) BOOL **statePaperFeed;**`

Return value

YES	Feeding
NO	Stopped

stateReturnWaiting — Retrieve return waiting status

Retrieves the return waiting status.

Syntax `@property(readonly) BOOL **stateReturnWaiting;**`

Return value

YES	In return waiting status
NO	Not in return waiting status

stateFlashMemoryRewriting — Retrieve FLASH memory rewrite status

Retrieves the FLASH memory rewrite status.

Syntax `@property(readonly) BOOL **stateFlashMemoryRewriting;**`

Return value

YES	In rewrite status
NO	Not in rewrite status

stateBattery — Retrieve battery voltage status

Retrieves the battery voltage status.

Syntax `@property(readonly) NSInteger **stateBattery;**`

Return value

0	No battery
1	Battery remaining capacity level 4 or 5 (Need to charge or battery remaining capacity is 0%)
2	Battery remaining capacity level 3 (battery remaining capacity: approx. 10%)
3	Battery remaining capacity level 2 (battery remaining capacity: approx. 40%)
4	Battery remaining capacity level 1 (battery remaining capacity: approx. 80%)

(3) **SIIMpaPrinterDiscovery**

- **Summary**

This class provides printer search functions.
This class provides the following functions.

Public Methods

Method	Function Summary
startDiscoveryPrinter	Start printer search
cancelDiscoveryPrinter	Cancel printer search
getFoundPrinter	Retrieve information list of searched printer

Constants used when retrieving the events are shown in the following table.

Constants

Method	Function Summary
SII_MPM_EVENT_FINISHED_DISCOVERY	Finish printer search Value = 0
SII_MPM_EVENT_CANCELED_DISCOVERY	Cancel printer search Value = 1

- **Public Methods**

startDiscoveryPrinter — Start printer search

Starts the printer search.

Syntax

```
- (BOOL) startDiscoveryPrinter: (SEL) method
                    target: (NSObject*) target
                    prnIf: (PrinterInterface) prnIf
                    retry: (NSInteger) retry
                    timeout: (NSInteger) timeout
                    error: (NSError**) error;
```

Parameter

method — Specification of selector
Specify the selector of method to execute when the printer search is finished or the search is canceled by **cancelDiscoveryPrinter**.
The sample of execution method by the callback is follows.

```
- (void)onDiscoveryFinished:(NSNumber *)event;
```

When this method is called by search finish, **SII_MPM_EVENT_FINISHED_DISCOVERY** is assigned to `event`.
When this method is called by search cancel, **SII_MPM_EVENT_CANCELED_DISCOVERY** is assigned to `event`.

target — Specification of object
Specify the object including the method specified by **method**.

prnIf — Constants of communication interface
See "4.4.2(1) **PrinterInterface**" for the available setting.

retry Number of retry times (time)
The performance varies depending on the setting of **prnIf**.
・When **SII_MPM_PRN_IF_TCP** is specified:
Sends local broadcast packet as much as the specified times in **retry**.
The range is 1 to 5.
When the specified value is below 1, the value is processed as 1.
When the specified value exceeds 5, the value is processed as 5.
・When **SII_MPM_PRN_IF_BT** is specified:
The specified value is ignored, and the search is performed 1 time only.

timeout Timeout period for 1 time search
The performance varies depending on the setting of **prnIf**.
・When **SII_MPM_PRN_IF_TCP** is specified:
Every time of sending local broadcast packet, wait for a response from the printer until the time specified by **timeout**.
The range is 3 to 60.
When the specified value is below 3, the value is processed as 3.
When the specified value exceeds 60, the value is processed as 60.
・When **SII_MPM_PRN_IF_BT** is specified:
The specified value is ignored.

error Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description Cancel is available by **cancelDiscoveryPrinter**.

Retrieve the search result by **getFoundPrinter**.

cancelDiscoveryPrinter Cancel printer search

Cancels the printer search started by **startDiscoveryPrinter**.

Syntax `- (BOOL) cancelDiscoveryPrinter: (NSError**) error`

Parameter **error** Error object reference

Return value The following value may be thrown when an error occurs while calling this method.
When an error occurs, see **NSError** object specified by **error.**

YES Normal
NO Error

Description When the search is canceled, the method specified in **method** of **startDiscoveryPrinter** is executed.

The search result by cancellation can be retrieved by **getFoundPrinter**.

getFoundPrinter — Retrieve information list of searched printer

Retrieves all the printer information searched by **`startDiscoveryPrinter`**.

Syntax

```
- (BOOL) getFoundPrinter: (NSMutableArray**) foundPrinter
                   error: (NSError**) error
```

Parameter

`error`	Error object reference

Return value

The following value may be thrown when an error occurs while calling this method. When an error occurs, see **`NSError`** object specified by **`error.`**

YES	Normal
NO	Error

Description

Retrieves the searched printer information by **`NSMutableArray`**. Stores each of the printer information in **`SIIMpaDiscoveryPrinter`** class. By property of **`SIIMpaDiscoveredPrinter`** class, the printer model name, Bluetooth address, IP address, MAC address and serial number can be retrieved.
For the property of **`SIIMpaDiscoveredPrinter`** class, see "4.4.1(4) **`SIIMpaDiscoveredPrinter`**".

(4) **SIIMpaDiscoveredPrinter**

- **Summary**

  This class stores the printer information searched by **startDiscoveryPrinter**. By the printer information searched, the printer model name, Bluetooth address, IP address, MAC address and serial number can be retrieved.
  This class provides the following functions.

  Public Properties

Property	Function Summary
printerModel	Retrieve printer model name
bluetoothAddress	Retrieve Bluetooth address
ipAddress	Retrieve IP address
macAddress	Retrieve MAC address
serialNumber	Retrieve serial number

- **Public Properties**

printerModel — Retrieve printer model name

Retrieves the character string of printer model name.

Syntax `@property(readonly) NSString *printerModel;`

Return value Printer model name

bluetoothAddress — Retrieve Bluetooth address

Retrieves the character string of Bluetooth address.

Syntax `@property(readonly) NSString *bluetoothAddress;`

Return value Bluetooth address

ipAddress — Retrieve IP address

Retrieves the character string of IP address.

Syntax `@property(readonly) NSString *ipAddress;`

Return value IP address

macAddress — Retrieve MAC address

Retrieves the character string of MAC address.

Syntax `@property(readonly) NSString *macAddress;`

Return value MAC address

serialNumber — Retrieve Serial Number

Retrieves the character string of Serial Number.

Syntax `@property (readonly) NSString *serialNumber;`

Return value Serial Number

4.4.2 Enumerated Type (Enum)

(1) **`PrinterInterface`**

Enumerator used for specifying the communication interface of **`open`**.

Constant Name	Description
SII_MPM_PRN_IF_TCP	TCP/IP
SII_MPM_PRN_IF_BT	Bluetooth

(2) **`PrinterModel`**

Enumerator used for specifying the printer model of **`open`**.

Constant Name	Description
SII_MPM_PRN_MODEL_MP_A40	MP-A40

(3) **`Dithering`**

Enumerator used for dithering of **`resisterLogo`**, **`sendDataFile`**, and **`printPDF`**.

Constant Name	Description
SII_MPM_DITHERING_DISABLE	Dithering is disabled
SII_MPM_DITHERING_ERRORDIFFUSION	Dithering is enabled

(4) **`PrinterInformation`**

Enumerator used for specifying the printer information of **`getPrinterInformation`**, **`getPrinterInformationNumber`** and **`getPrinterInformationString`**.
See "MP-A40 THERMAL PRINTER TECHNICAL REFERENCE" for details of printer information to retrieve.

Constant Name	Description (Printer Information)
SII_MPM_GET_NV_MEM_CAP	NV graphics memory capacity
SII_MPM_GET_NV_MEM_REM_CAP	NV graphics memory remaining capacity
SII_MPM_GET_NV_MEM_KEYCODE_LIST	NV graphics key code list
SII_MPM_GET_REM_USER_MEM_CAP_DEFRAG	Remaining user area after defragment
SII_MPM_GET_REM_USER_MEM_CAP	Remaining user area
SII_MPM_GET_FUNC_SET_RESP	Function setting response
SII_MPM_GET_PRN_ID_MODEL	Printer ID send (Model ID)
SII_MPM_GET_PRN_ID_TYPE	Printer ID send (Type ID)
SII_MPM_GET_PRN_ID_ROM_VER	Printer ID send (ROM version ID)
SII_MPM_GET_PRN_ID_FIRM_VER_MAIN	Printer ID send (Firmware version (main))
SII_MPM_GET_PRN_ID_MFR	Printer ID send (Manufacturer)
SII_MPM_GET_PRN_ID_MODEL_NAME	Printer ID send (Model name)
SII_MPM_GET_PRN_ID_FIRM_VER_BOOT	Printer ID send (Firmware version (boot))
SII_MPM_GET_PRN_ID_FIRM_CHECKSUM_BOOT	Printer ID send (Firmware checksum (boot))

Constant Name	Description (Printer Information)
SII_MPM_GET_PRN_ID_FIRM_CHECKSUM_MAIN	Printer ID send (Firmware checksum (main))
SII_MPM_GET_PRN_ID_FIRM_CHECKSUM	Printer ID send (Firmware checksum (main+boot))
SII_MPM_GET_MAINT_NUM_FEED_LINE	Maintenance counter (Paper feed line count (in 100 dot-lines))
SII_MPM_GET_MAINT_NUM_HEAD_ACTIVE	Maintenance counter (Number of thermal head activation times (in 100 dot-lines))
SII_MPM_GET_MAINT_DRIVE_TIME	Maintenance counter (Drive time of printer mechanism (minute))
SII_MPM_GET_MAINT_NUM_FEED_LINE_ INTEGRATION	Maintenance counter (Paper feed line count (in 100 dot-lines) (integrated value))
SII_MPM_GET_MAINT_ NUM_HEAD_ACTIVE_INTEGRATION	Maintenance counter (Number of thermal head activation times (in 100 dot-lines) (integration))
SII_MPM_GET_MAINT_DRIVE_TIME_INTEGRATION	Maintenance counter (Drive time of printer mechanism (minute) (integration))
SII_MPM_GET_HFONT_24_CHECKSUM	1-byte font ID send (24 dots font, checksum)
SII_MPM_GET_HFONT_24_ID	1-byte font ID send (24 dots font, ID)
SII_MPM_GET_HFONT_24_INT_CHAR	1-byte font ID send (24 dots font, registered international character)
SII_MPM_GET_HFONT_16_CHECKSUM	1-byte font ID send (16 dots font, checksum)
SII_MPM_GET_HFONT_16_ID	1-byte font ID send (16 dots font, ID)
SII_MPM_GET_HFONT_16_INT_CHAR	1-byte font ID send (16 dots font, registered international character)
SII_MPM_GET_FFONT_LANG	2-byte font ID send (Language)
SII_MPM_GET_FFONT_STANDARD	2-byte font ID send (Standard)
SII_MPM_GET_FFONT_COMPANY	2-byte font ID send (Company name)
SII_MPM_GET_FFONT_CHECKSUM	2-byte font ID send (Checksum)
SII_MPM_GET_FFONT_DATA_SIZE	2-byte font ID send (Data size)
SII_MPM_GET_FFONT_ROM_ID	2-byte font ID send (ROMID)

(5) **`TransactionFunction`**

Enumerator used for specifying the batch process control method of **`controlTransaction`**.

Constant Name	Description
SII_MPM_TRANSACTION_CLEAR	Interrupt batch process
SII_MPM_TRANSACTION_START	Begin batch process
SII_MPM_TRANSACTION_PRINT	End batch print and batch process

(6) **`Direction`**

Enumerator used for specifying the print direction of **`selectPageMode`**. The default value is shown by the shaded enumerated value in the following table.
When the start point is "top left" or "bottom right", the printer maps the print data in a vertical direction to the paper feed direction.
When the start point is "top right" or "bottom left", the printer maps the print data in a parallel direction to the paper feed direction.

Constant Name	Description
SII_MPM_DIRECTION_LEFT_TO_RIGHT	Print direction: left to right, Start point: top left (Figure 4-2 A)
SII_MPM_DIRECTION_BOTTOM_TO_TOP	Print direction: bottom to top, Start point: bottom left (Figure 4-2 B)
SII_MPM_DIRECTION_RIGHT_TO_LEFT	Print direction: right to left, Start point: bottom right (Figure 4-2 C)
SII_MPM_DIRECTION_TOP_TO_BOTTOM	Print direction: top to bottom, Start point: top right (Figure 4-2 D)

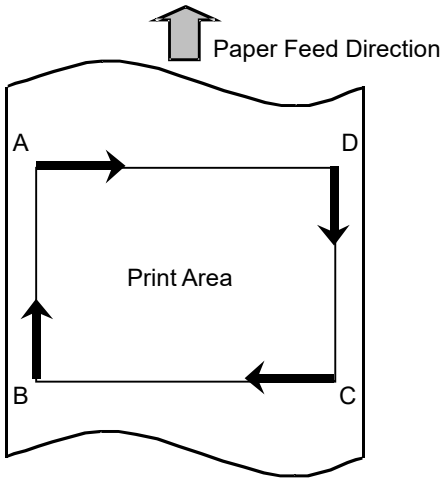


Figure 4-2

(7) **`Alignment`**

Enumerator used for specifying the print position of **`setStandardModeAlignment`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_ALIGN_LEFT	Left aligned
SII_MPM_ALIGN_CENTER	Centered
SII_MPM_ALIGN_RIGHT	Right aligned

(8) **`CharacterSet`**

Enumerator used for specifying the character set of **`selectCharacterSet`** and **`getCharacterSet`**. The default value varies depending on the language setting of iOS device.

Japanese: **SII_MPM_CODEPAGE_KATAKANA**
Other than Japanese: **SII_MPM_CODEPAGE_1252**

Constant Name	Description
SII_MPM_CODEPAGE_437	USA, Standard Europe (Code Page 437)
SII_MPM_CODEPAGE_KATAKANA	Katakana
SII_MPM_CODEPAGE_850	Multilingual (Code Page 850)
SII_MPM_CODEPAGE_860	Portuguese (Code Page 860)
SII_MPM_CODEPAGE_863	Canadian-French (Code Page 863)
SII_MPM_CODEPAGE_865	Nordic (Code Page 865)
SII_MPM_CODEPAGE_857[*1]	Turkish (Code Page 857)
SII_MPM_CODEPAGE_737	Greek (Code Page 737)
SII_MPM_CODEPAGE_1252	Latin (Code Page 1252)
SII_MPM_CODEPAGE_866	Russian (Code Page 866)
SII_MPM_CODEPAGE_852	Eastern Europe (Code Page 852)
SII_MPM_CODEPAGE_858	Euro (Code Page 858)
SII_MPM_CODEPAGE_855	Cyrillic (Code Page 855)
SII_MPM_CODEPAGE_864[*1]	Arabic (Code Page 864)
SII_MPM_CODEPAGE_1250	Central European (Code Page 1250)
SII_MPM_CODEPAGE_1251	Cyrillic (Code Page 1251)
SII_MPM_CODEPAGE_1253[*2]	Greek (Code Page 1253)
SII_MPM_CODEPAGE_1254	Turkish (Code Page 1254)

*1: 20ACh of the Unicode cannot be printed.
*2: 00AAh of the Unicode cannot be printed.

(9) **`InternationalCharacterSet`**

Enumerator used for specifying the international character set of **`selectInternationalCharacterSet`** and **`getInternationalCharacter.`**
The default value varies depending on the language setting of iOS device.

Japanese: **SII_MPM_INT_CHAR_SET_JAPAN**
Other than Japanese: **SII_MPM_INT_CHAR_SET_USA**

Constant Name	Description
SII_MPM_INT_CHAR_SET_USA	USA
SII_MPM_INT_CHAR_SET_FRANCE	France
SII_MPM_INT_CHAR_SET_GERMANY	Germany
SII_MPM_INT_CHAR_SET_UNITED_KINGDOM	United Kingdom
SII_MPM_INT_CHAR_SET_DENMARK_1	Denmark
SII_MPM_INT_CHAR_SET_SWEDEN	Sweden
SII_MPM_INT_CHAR_SET_ITALY	Italy
SII_MPM_INT_CHAR_SET_SPAIN_1	Spain I
SII_MPM_INT_CHAR_SET_JAPAN	Japan
SII_MPM_INT_CHAR_SET_NORWAY	Norway
SII_MPM_INT_CHAR_SET_DENMARK_2	Denmark II
SII_MPM_INT_CHAR_SET_SPAIN_2	Spain II
SII_MPM_INT_CHAR_SET_LATIN_AMERICA	Latin America
SII_MPM_INT_CHAR_SET_ARABIA	Arabia

(10) **`CharacterType`**

Enumerator used for specifying the character font of **`setCharacterFormatting`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_CHAR_TYPE_FONT_CURRENT[*1]	Current Setting (without sending the setting printer command)
SII_MPM_CHAR_TYPE_FONT_A	Font A (24 × 12), Kanji font A (24 × 24)
SII_MPM_CHAR_TYPE_FONT_B[*1]	Font B (16 × 8), Kanji font B (16 × 16)

*1: When **SII_MPM_CODEPAGE_864** is selected in **`selectCharacterSet`**, the text is printed in FontA (24 × 12) regardless of specifying the character font in **`setCharacterFormatting`**.

(11) **`CharacterScale`**

Enumerator used for specifying the character scale of **`setCharacterFormatting`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_CHAR_SCALE_CURRENT	Current Setting (without sending the setting printer command)
SII_MPM_CHAR_SCALE_X1	× 1 (Standard)
SII_MPM_CHAR_SCALE_X2	× 2 (double)
SII_MPM_CHAR_SCALE_X3	× 3 (triple)
SII_MPM_CHAR_SCALE_X4	× 4 (quadruple)
SII_MPM_CHAR_SCALE_X5	× 5 (quintuple)
SII_MPM_CHAR_SCALE_X6	× 6 (sextuple)
SII_MPM_CHAR_SCALE_X7	× 7 (septuple)
SII_MPM_CHAR_SCALE_X8	× 8 (octuple)

(12) **`Underline`**

Enumerator used for specifying the underline of **`setCharacterFormatting`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_CHAR_UNDERLINE_ CURRENT	Current Setting (without sending the setting printer command)
SII_MPM_CHAR_UNDERLINE_NONE	No underline
SII_MPM_CHAR_UNDERLINE_1DOT	Specify 1 dot width underline
SII_MPM_CHAR_UNDERLINE_2DOT	Specify 2 dots width underline

(13) **`Bold`**

Enumerator used for specifying the bold print of **`setCharacterFormatting`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_CHAR_BOLD_CURRENT	Current Setting (without sending the setting printer command)
SII_MPM_CHAR_BOLD_OFF	No bold print
SII_MPM_CHAR_BOLD_ON	Specify bold print

(14) **`Reverse`**

Enumerator used for specifying the reverse print of **`setCharacterFormatting`**. The default value is shown by the shaded enumerated value in the following table.

Constant Name	Description
SII_MPM_CHAR_REVERSE_ CURRENT	Current Setting (without sending the setting printer command)
SII_MPM_CHAR_REVERSE_OFF	No reverse print
SII_MPM_CHAR_REVERSE_ON	Specify reverse print

(15) **`Rotate`**

Enumerator used for specifying the rotation print of **`setCharacterFormatting`**, for specifying print direction of **`setStandardModeBarcodeDirection`**, and for specifying the image rotation direction of **`printPDF`**. The default value is shown by the shaded enumerated value in the following table.
When the character rotation print is specified **`setCharacterFormatting`**, the print position is changed. The print position and direction of text is shown in the Figure 4-3, Figure 4-4, Figure 4-5 and Figure 4-6.

Constant Name	Description
SII_MPM_ROTATE_CURRENT[*1]	Current Setting (without sending the setting printer command)
SII_MPM_ROTATE_NONE	No rotation
SII_MPM_ROTATE_90_TO_RIGHT[*1]	Rotate 90 degrees to right
SII_MPM_ROTATE_180	Rotate 180 degrees
SII_MPM_ROTATE_90_TO_LEFT[*1]	Rotate 90 degrees to left

*1: Not supported in **`printPDF`**.

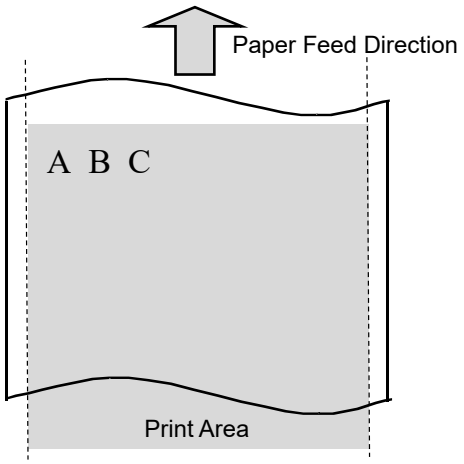


Figure 4-3 No rotation

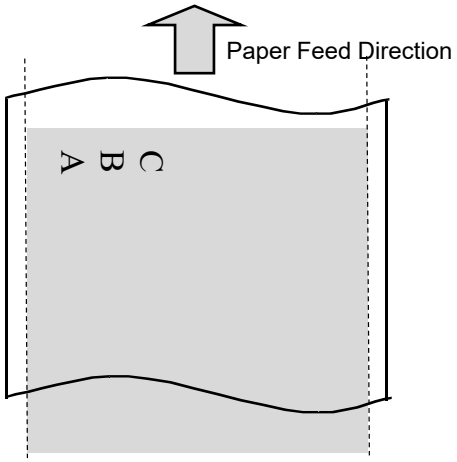


Figure 4-4 Rotate 90 degrees to right

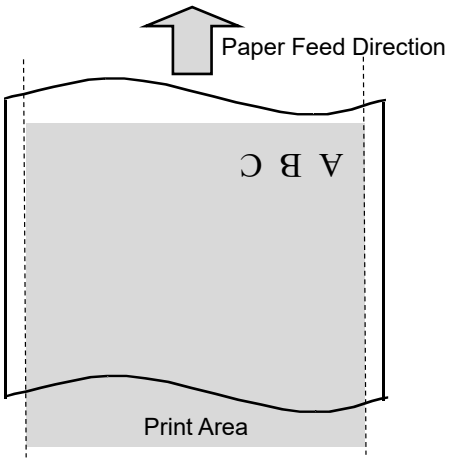


Figure 4-5 Rotate 180 degrees

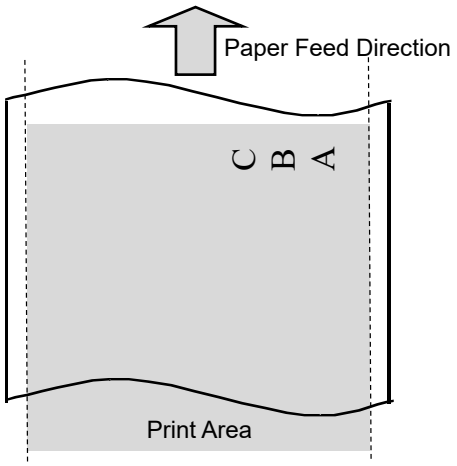


Figure 4-6 Rotate 90 degrees to left

(16) **`TypeBarcode`**

Enumerator used for specifying the barcode type of **`printBarcode`**.

Constant Name	Description	
SII_MPM_TYPE_BARCODE_UPC_A	UPC-A	Multilevel barcode
SII_MPM_TYPE_BARCODE_UPC_E	UPC-E	Multilevel barcode
SII_MPM_TYPE_BARCODE_JAN13	JAN13 (EAN13)	Multilevel barcode
SII_MPM_TYPE_BARCODE_JAN8	JAN8 (EAN8)	Multilevel barcode
SII_MPM_TYPE_BARCODE_CODE39	CODE39	Binary level barcode
SII_MPM_TYPE_BARCODE_ITF	ITF	Binary level barcode
SII_MPM_TYPE_BARCODE_CODABAR	CODABAR	Binary level barcode
SII_MPM_TYPE_BARCODE_CODE128	CODE128	Multilevel barcode
SII_MPM_TYPE_BARCODE_CODE93	CODE93	Multilevel barcode
SII_MPM_TYPE_BARCODE_JAN13_ADDON2	JAN13 (EAN13) add-on 2	Multilevel barcode
SII_MPM_TYPE_BARCODE_JAN13_ADDON5	JAN13 (EAN13) add-on 5	Multilevel barcode
SII_MPM_TYPE_BARCODE_GS1_OMNI_ DIRECTIONAL	GS1 Databar Omni-directional	Multilevel barcode
SII_MPM_TYPE_BARCODE_GS1_TRUNCATED	GS1 Databar Truncated	Multilevel barcode
SII_MPM_TYPE_BARCODE_GS1_LIMITED	GS1 Databar Limited	Multilevel barcode
SII_MPM_TYPE_BARCODE_GS1_EXPANDED	GS1 Databar Expanded	Multilevel barcode

(17) **`ModuleWidthBarcode`**

Enumerator used for specifying the barcode module width or narrow element of **`printBarcode.`**

Constant Name	Description
SII_MPM_MODULE_WIDTH_BARCODE_2	2 dots
SII_MPM_MODULE_WIDTH_BARCODE_3	3 dots
SII_MPM_MODULE_WIDTH_BARCODE_4	4 dots
SII_MPM_MODULE_WIDTH_BARCODE_5	5 dots
SII_MPM_MODULE_WIDTH_BARCODE_6	6 dots

(18) **HriPositionBarcode**

Enumerator used for specifying the barcode HRI characters of **printBarcode**.

Constant Name	Description
SII_MPM_HRI_BARCODE_NONE	No HRI character
SII_MPM_HRI_BARCODE_TOP_FONT_A	Above barcode (font A)
SII_MPM_HRI_BARCODE_BOTTOM_FONT_A	Below barcode (font A)
SII_MPM_HRI_BARCODE_FONT_A	Above and below barcode (font A)
SII_MPM_HRI_BARCODE_TOP_FONT_B	Above barcode (font B)
SII_MPM_HRI_BARCODE_BOTTOM_FONT_B	Below barcode (font B)
SII_MPM_HRI_BARCODE_FONT_B	Above and below barcode (font B)

(19) **NwRatioBarcode**

Enumerator used for specifying the barcode N:W ratio of **printBarcode**.

Constant Name	Description
SII_MPM_NWRATIO_BARCODE_1TO2	1:2
SII_MPM_NWRATIO_BARCODE_1TO2_5	1:2.5
SII_MPM_NWRATIO_BARCODE_1TO3	1:3

(20) **Type2DCode**

Enumerator used for specifying the 2-dimensional barcode type of **print2DCode**.

Constant Name	Description
SII_MPM_TYPE_2DCODE_QR_CODE	QR Code
SII_MPM_TYPE_2DCODE_PDF417	PDF417
SII_MPM_TYPE_2DCODE_DATA_MATRIX	Data Matrix
SII_MPM_TYPE_2DCODE_MAXI_CODE	Maxi Code
SII_MPM_TYPE_2DCODE_GS1_STACKED	GS1 Databar Stacked
SII_MPM_TYPE_2DCODE_GS1_OMNI_DIRECTIONAL	GS1 Databar Stacked Omni-directional
SII_MPM_TYPE_2DCODE_GS1_EXPANDED_STACKED	GS1 Databar Expanded Stacked

(21) **Mode2Dcode**

Enumerator used for specifying the 2-dimensional barcode mode of **print2DCode**.

Constant Name	Description	
SII_MPM_MODE_2DCODE_QR_CODE_MODEL1	QR Code	Model1
SII_MPM_MODE_2DCODE_QR_CODE_MODEL2	QR Code	Model2
SII_MPM_MODE_2DCODE_PDF417_STANDARD	PDF417	Normal Mode
SII_MPM_MODE_2DCODE_PDF417_COMPACT	PDF417	Simple Mode
SII_MPM_MODE_2DCODE_DATA_MATRIX_AUTO	Data Matrix	Module numbers: Automatic
SII_MPM_MODE_2DCODE_DATA_MATRIX_10_10	Data Matrix	Module numbers: 10 × 10

Constant Name	Description	
SII_MPM_MODE_2DCODE_DATA_MATRIX_12_12	Data Matrix	Module numbers: 12 × 12
SII_MPM_MODE_2DCODE_DATA_MATRIX_14_14	Data Matrix	Module numbers: 14 × 14
SII_MPM_MODE_2DCODE_DATA_MATRIX_16_16	Data Matrix	Module numbers: 16 × 16
SII_MPM_MODE_2DCODE_DATA_MATRIX_18_18	Data Matrix	Module numbers: 18 × 18
SII_MPM_MODE_2DCODE_DATA_MATRIX_20_20	Data Matrix	Module numbers: 20 × 20
SII_MPM_MODE_2DCODE_DATA_MATRIX_22_22	Data Matrix	Module numbers: 22 × 22
SII_MPM_MODE_2DCODE_DATA_MATRIX_24_24	Data Matrix	Module numbers: 24 × 24
SII_MPM_MODE_2DCODE_DATA_MATRIX_26_26	Data Matrix	Module numbers: 26 × 26
SII_MPM_MODE_2DCODE_DATA_MATRIX_32_32	Data Matrix	Module numbers: 32 × 32
SII_MPM_MODE_2DCODE_DATA_MATRIX_36_36	Data Matrix	Module numbers: 36 × 36
SII_MPM_MODE_2DCODE_DATA_MATRIX_40_40	Data Matrix	Module numbers: 40 × 40
SII_MPM_MODE_2DCODE_DATA_MATRIX_44_44	Data Matrix	Module numbers: 44 × 44
SII_MPM_MODE_2DCODE_DATA_MATRIX_48_48	Data Matrix	Module numbers: 48 × 48
SII_MPM_MODE_2DCODE_DATA_MATRIX_52_52	Data Matrix	Module numbers: 52 × 52
M SII_MPM_ODE_2DCODE_DATA_MATRIX_64_64	Data Matrix	Module numbers: 64 × 64
SII_MPM_MODE_2DCODE_DATA_MATRIX_72_72	Data Matrix	Module numbers: 72 × 72
SII_MPM_MODE_2DCODE_DATA_MATRIX_80_80	Data Matrix	Module numbers: 80 × 80
SII_MPM_MODE_2DCODE_DATA_MATRIX_88_88	Data Matrix	Module numbers: 88 × 88
SII_MPM_MODE_2DCODE_DATA_MATRIX_96_96	Data Matrix	Module numbers: 96 × 96
SII_MPM_MODE_2DCODE_DATA_MATRIX_104_104	Data Matrix	Module numbers: 104 × 104
SII_MPM_MODE_2DCODE_DATA_MATRIX_120_120	Data Matrix	Module numbers: 120 × 120
SII_MPM_MODE_2DCODE_DATA_MATRIX_132_132	Data Matrix	Module numbers: 132 × 132
SII_MPM_MODE_2DCODE_DATA_MATRIX_144_144	Data Matrix	Module numbers: 144 × 144
SII_MPM_MODE_2DCODE_DATA_MATRIX_8_18	Data Matrix	Module numbers: 8 × 18
SII_MPM_MODE_2DCODE_DATA_MATRIX_8_32	Data Matrix	Module numbers: 8 × 32
SII_MPM_MODE_2DCODE_DATA_MATRIX_12_26	Data Matrix	Module numbers: 12 × 26
SII_MPM_MODE_2DCODE_DATA_MATRIX_12_36	Data Matrix	Module numbers: 12 × 36
SII_MPM_MODE_2DCODE_DATA_MATRIX_16_36	Data Matrix	Module numbers: 16 × 36
SII_MPM_MODE_2DCODE_DATA_MATRIX_16_48	Data Matrix	Module numbers: 16 × 48
SII_MPM_MODE_2DCODE_MAXI_CODE_2	Maxi Code	Mode2
SII_MPM_MODE_2DCODE_MAXI_CODE_3	Maxi Code	Mode3
SII_MPM_MODE_2DCODE_MAXI_CODE_4	Maxi Code	Mode4
SII_MPM_MODE_2DCODE_MAXI_CODE_5	Maxi Code	Mode5
SII_MPM_MODE_2DCODE_NONE	GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(22) **`ModuleSize2DCode`**

Enumerator used for specifying the 2-dimensional barcode module size of **`print2DCode.`**

Constant Name	Description	
SII_MPM_MODULE_SIZE_2DCODE_ DEFAULT	QR Code	Default (6 dots)
	PDF417	Default (4 dots)
	Data Matrix	Default (6 dots)
	Maxi Code	No settings
	GS1 Databar Stacked	Default (6 dots)
	GS1 Databar Stacked Omni-directional	Default (6 dots)
	GS1 Databar Expanded Stacked	Default (6 dots)
SII_MPM_MODULE_SIZE_2DCODE_2	QR Code	2 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_3	QR Code	3 dots
	PDF417	
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_4	QR Code	4 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_5	QR Code	5 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_6	QR Code	6 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Constant Name	Description	
SII_MPM_MODULE_SIZE_2DCODE_7	QR Code	7 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_8	QR Code	8 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_9	QR Code	9 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_10	QR Code	10 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_11	QR Code	11 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_12	QR Code	12 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_13	QR Code	13 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

Constant Name	Description	
SII_MPM_MODULE_SIZE_2DCODE_14	QR Code	14 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_15	QR Code	15 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	
SII_MPM_MODULE_SIZE_2DCODE_16	QR Code	16 dots
	Data Matrix	
	GS1 Databar Stacked	
	GS1 Databar Stacked Omni-directional	
	GS1 Databar Expanded Stacked	

(23) **`ErrorCorrect2DCode`**

Enumerator used for specifying the 2-dimensional barcode error correction level of **`print2DCode`**.

Constant Name	Description	
SII_MPM_ERR_CORRECTION_2DCODE_QR_CODE_L	QR Code	L
SII_MPM_ERR_CORRECTION_2DCODE_QR_CODE_M	QR Code	M
SII_MPM_ERR_CORRECTION_2DCODE_QR_CODE_Q	QR Code	Q
SII_MPM_ERR_CORRECTION_2DCODE_QR_CODE_H	QR Code	H
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_0	PDF417	0
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_1	PDF417	1
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_2	PDF417	2
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_3	PDF417	3
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_4	PDF417	4
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_5	PDF417	5
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_6	PDF417	6
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_7	PDF417	7
SII_MPM_ERR_CORRECTION_2DCODE_PDF417_8	PDF417	8
SII_MPM_ERR_CORRECTION_2DCODE_NONE	Data Matrix Maxi Code GS1 Databar Stacked GS1 Databar Stacked Omni-directional GS1 Databar Expanded Stacked	No settings

(24) **LogFileSize**

Enumerator used for specifying the maximum size of the log file of **setLog**.

Constant Name	Description
SII_MPM_LOG_FILE_SIZE_1MB	1 MB (1048576 bytes)
SII_MPM_LOG_FILE_SIZE_5MB	5 MB (5242880 bytes)
SII_MPM_LOG_FILE_SIZE_10MB	10 MB (10485760 bytes)
SII_MPM_LOG_FILE_SIZE_50MB	50 MB (52428800 bytes)

(25) **ErrorCode**

Enumerator used for retrieving the error code by **NSError**.

Constant Name	Description
SII_MPM_ERR_PARAM	Parameter is incorrect.
SII_MPM_ERR_OPENED	Specified printer has already been opened.
SII_MPM_ERR_NOT_OPENED	Specified printer is not opened.
SII_MPM_ERR_TIMEOUT	Timeout or busy state occurs.
SII_MPM_ERR_OFFLINE	Printer is disconnected or offline.
SII_MPM_ERR_CLOSE_FAIL	Failed to disconnect printer.
SII_MPM_ERR_NOT_MONITORING	Monitoring of connecting status is not performed.
SII_MPM_ERR_INIT_FAILED	Failed to initialize.
SII_MPM_ERR_DATA_SIZE_ZERO	0 byte size data is specified.
SII_MPM_ERR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.
SII_MPM_ERR_INVALID_DATA	Invalid data is specified.
SII_MPM_ERR_INVALID_STATE	Specified to access PrinterManager object for callback.
SII_MPM_ERR_ACCESS	Printer cannot be accessed.
SII_MPM_ERR_CANCELED	Function is canceled.
SII_MPM_ERR_WRITE_FAULT	Data cannot be sent to printer.
SII_MPM_ERR_FILE_INVALID	Specified file is invalid.
SII_MPM_ERR_ENCODE_FAILED	Error has occurred in encoding text data.
SII_MPM_ERR_NOT_FOUND	Specified file cannot be found.
SII_MPM_ERR_TRANSACTION_STOPPED	Batch process is not started.
SII_MPM_ERR_PRINTER_STATUS_ERROR	Printer status is abnormal.

(26) **ImageTransactionMode**

Enumerator used for specifying the image printing mode by **setImageTransactionMode**.

Constant Name	Description
SII_MPM_IMAGE_TRANSACTION_MODE_ON	A mode that prints in batch after mapping image data on the memory of the printer
SII_MPM_IMAGE_TRANSACTION_MODE_OFF	A mode that prints sequentially while receiving image data, without mapping it on the memory of the printer.

Chapter 5
Sample Program

This chapter describes the sample programs provided by the SDK.

5.1 Sample Program Overview

The SDK includes a sample program of XCode project format.

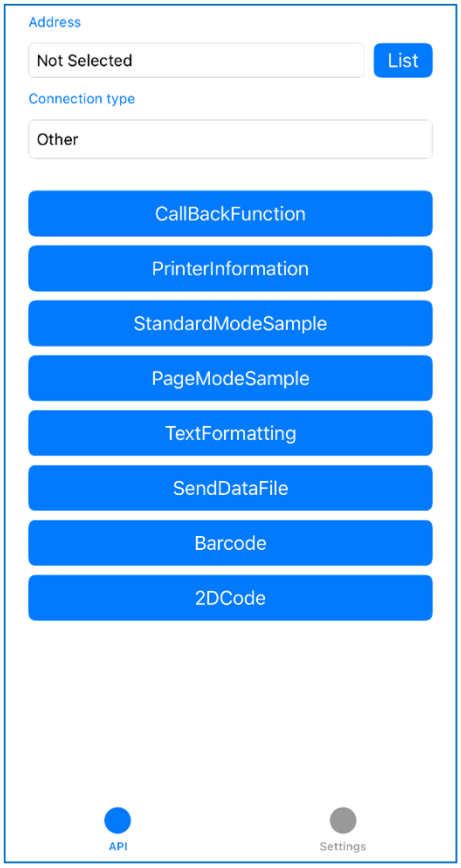


Figure 5-1

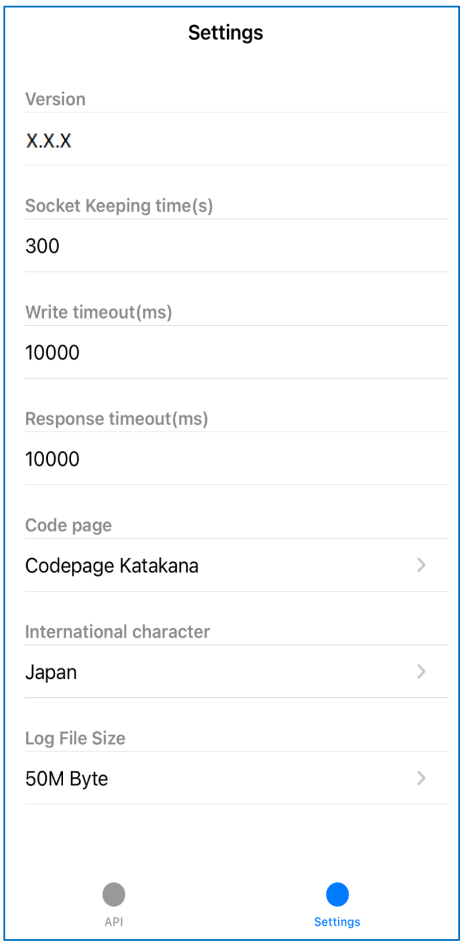


Figure 5-2

5.2 Sample Program Function

This section describes the functions of the sample program.

5.2.1 API Window

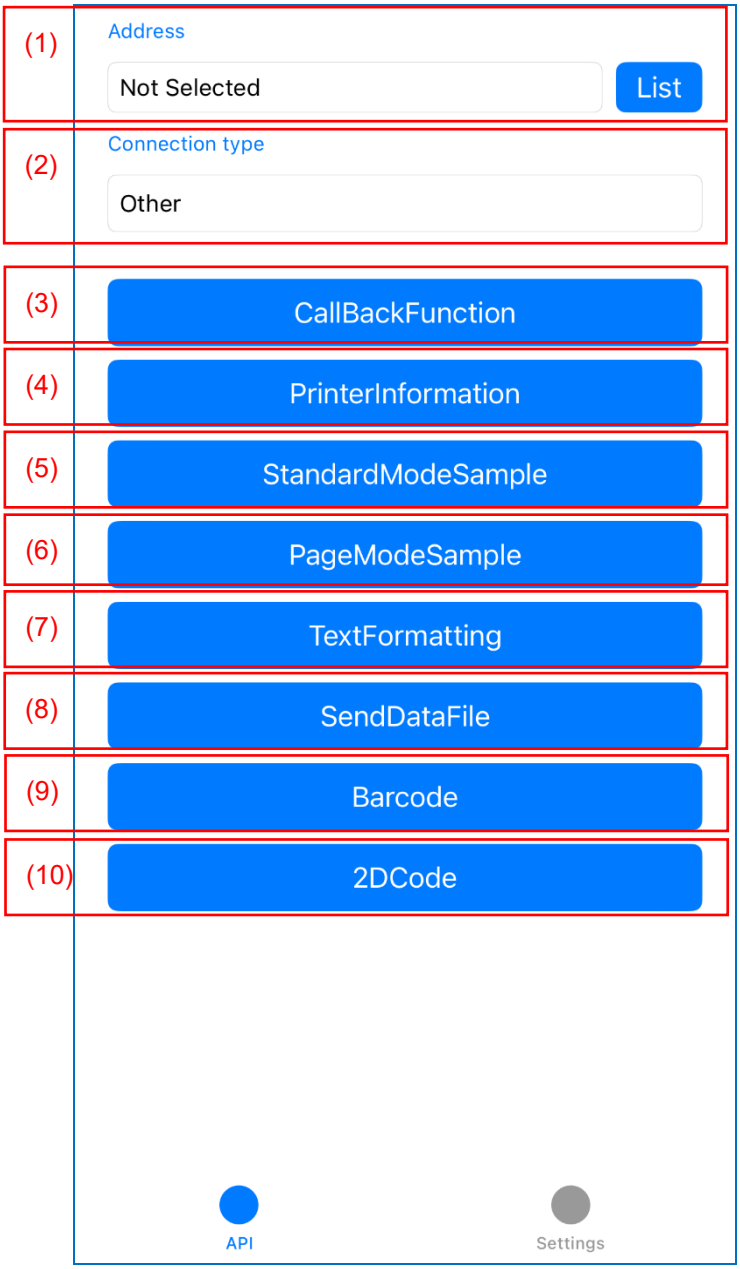


Figure 5-3

The functions of the sample program are as shown in Table 5-1.

Table 5-1 Function of Sample Program

No.	Description	Remarks
(1)	Address	Specifies printer address. By tapping [List] button, searches the Bluetooth device or TCP/IP device. The list of devices searched by **`startDiscoveryPrinter`** (Bluetooth) or **`startDiscoveryPrinter`** (TCP/IP) is displayed. By selecting a printer to connect from the list, Bluetooth address or IP address can be entered. Example: "00:11:22:AA:BB:CC", "192.168.0.190"
(2)	Connection type	Selects connection type to a printer.
(3)	CallbackFunction	Registers the callback of the printer status change in **`startCallbackFunction`**, and displays the change of status information in a dialog.
(4)	PrinterInformation	Retrieves the printer information from the printer by using **`getPrinterInformation`**, **`getPrinterInformationNumber`** and **`getPrinterInformationString`**.
(5)	StandardModeSample	In the standard mode, sends the sample receipt printing command created by various printing method to the printer.
(6)	PageModeSample	In the page mode, sends the sample receipt printing command created by various printing method to the printer.
(7)	TextFormatting	Sends the printing command formatted by various characters formatting to the printer.
(8)	SendDataFile	Sends the specified file by using **`sendDataFile`** to the printer.
(9)	Barcode	Sends the barcode printing command by using **`printBarcode`** to the printer.
(10)	2DCode	Sends the 2-dimensional barcode printing command by using **`print2Dcode`** to the printer.

5.2.2 Settings Window

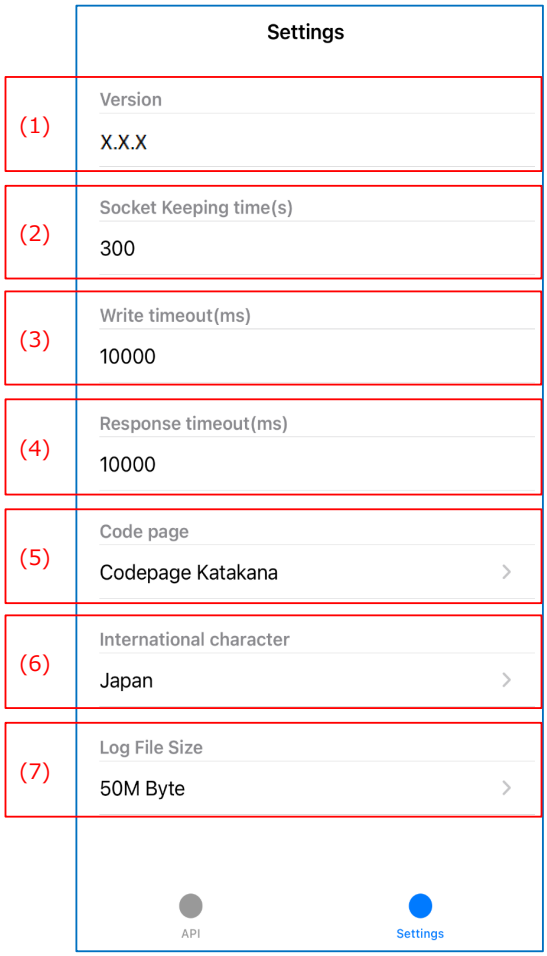


Figure 5-4

The functions of the sample program are as shown in Table 5-2.

Table 5-2 Settings Window

No.	Description	Remarks
(1)	Version	Displays the SDK version built in the sample program.
(2)	Socket Keeping time(s)	Sets the socket keeping time of TCP/IP connection in s (milliseconds). This setting value is used in **`socketKeepingTime`** which is the parameter of **`open`**.
(3)	Write timeout(ms)	Sets the time until the send timeout occurrence in ms (milliseconds).
(4)	Response timeout(ms)	Sets the time until the receive timeout occurrence in ms (milliseconds).
(5)	Code page	Sets the codepage. This is the encoding when printing texts (when text files are specified by TextFormatting or SendDataFile) and the character setting value for printer side.
(6)	International character	Sets the international character set. This is the international character setting value for printer side when printing texts (when text files are specified by TextFormatting or SendDataFile).
(7)	Log File Size	Sets the log file size to be stored by using **`setLog`**.

5.3 Precaution

No guarantee of proper operation and support are provided for sample programs.

Sample programs are subject to change without notice.

Chapter 6
Disclaimer

We closely monitor the development of this software in order to avoid problems. However, we are not responsible for any damages arising out of the use of this software.

Appendix A
Character Sets (Character Code Table)

A.1 Character Code Table

(When **SII_MPM_INT_CHAR_SET_USA** is set in the setting of international character set)

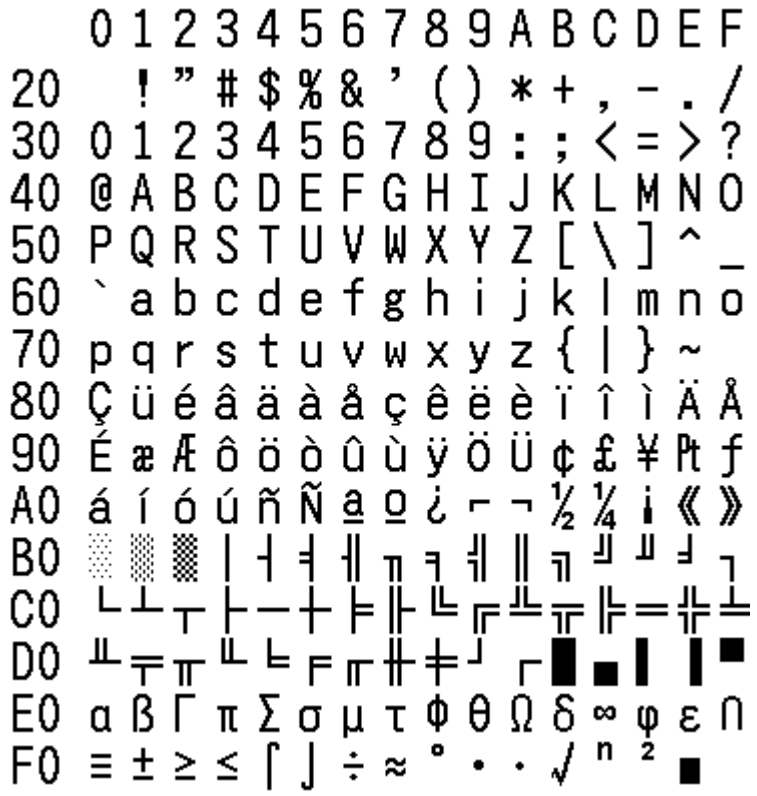


Figure A-1 SII_MPM_CODEPAGE_437 (USA, Standard Europe)

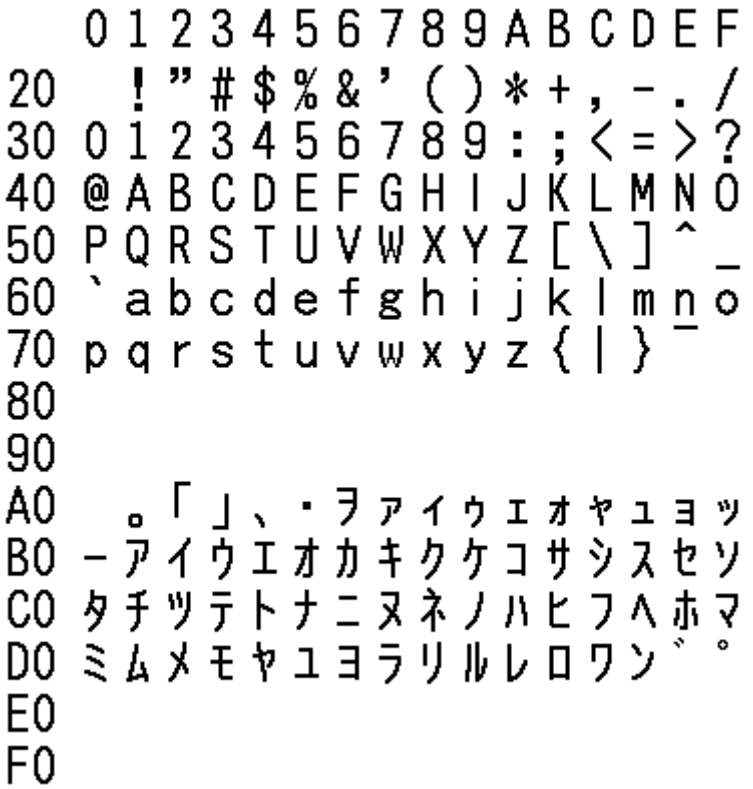


Figure A-2 SII_MPM_CODEPAGE_KATAKANA

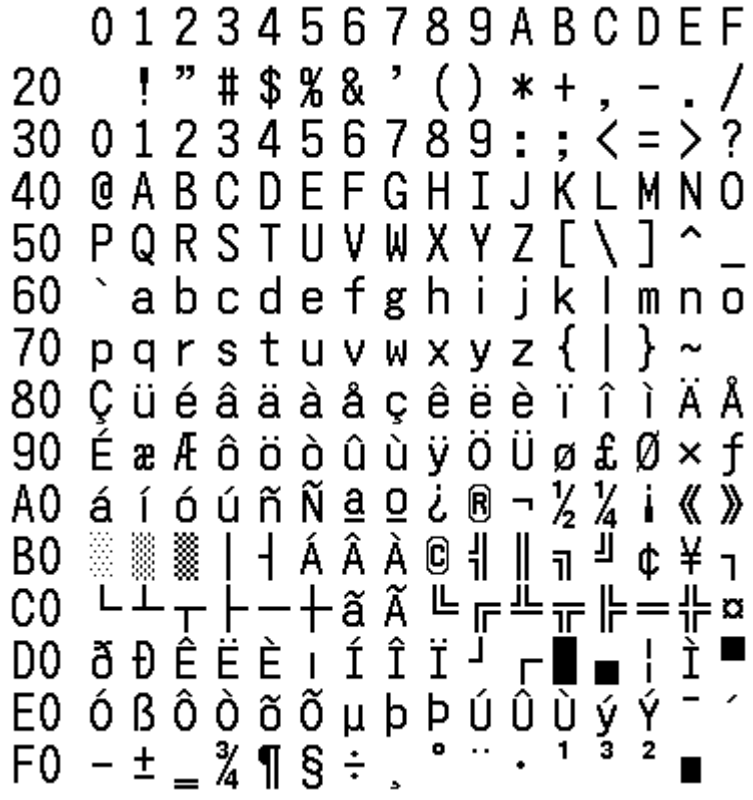


Figure A-3 SII_MPM_CODEPAGE_850 (Multilingual)

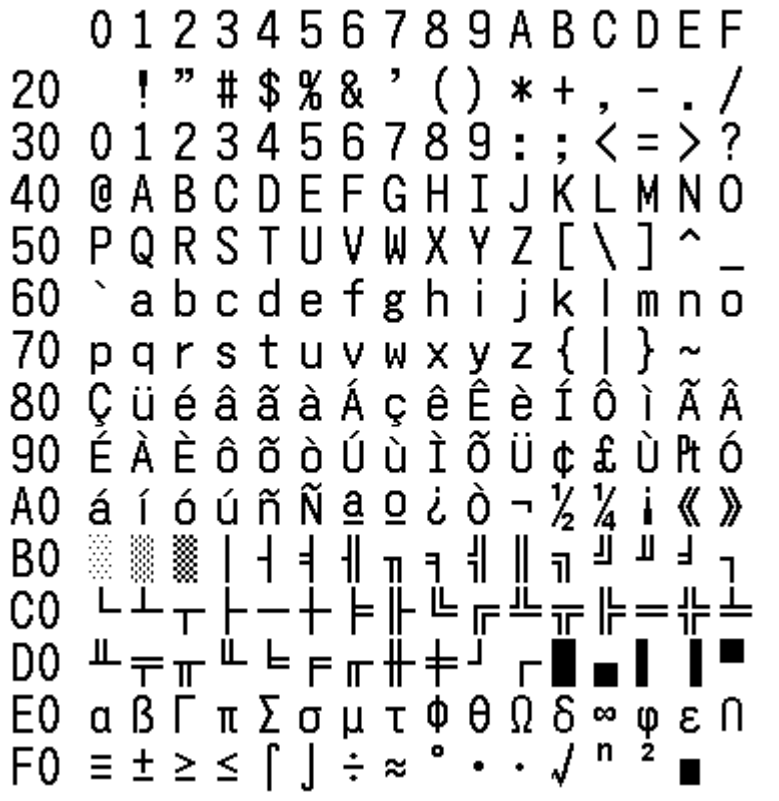


Figure A-4 SII_MPM_CODEPAGE_860 (Portuguese)

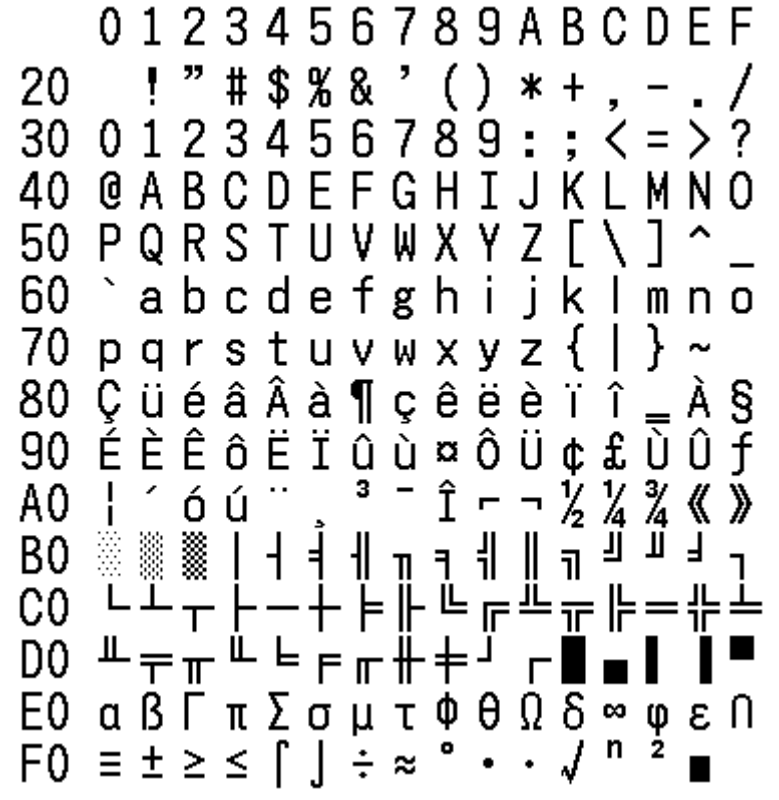


Figure A-5 SII_MPM_CODEPAGE_863 (Canadian-French)

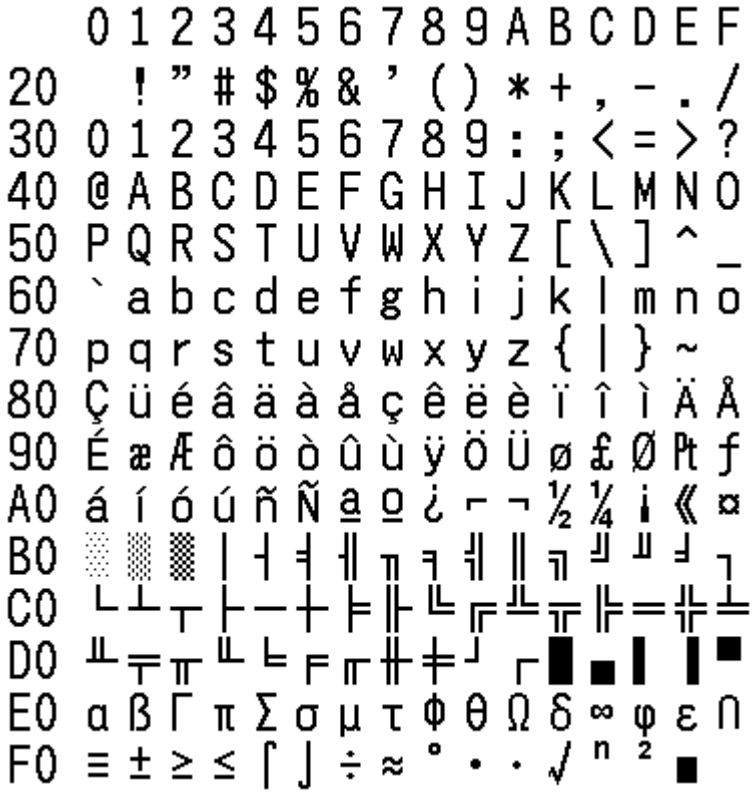


Figure A-6 SII_MPM_CODEPAGE_865 (Nordic)

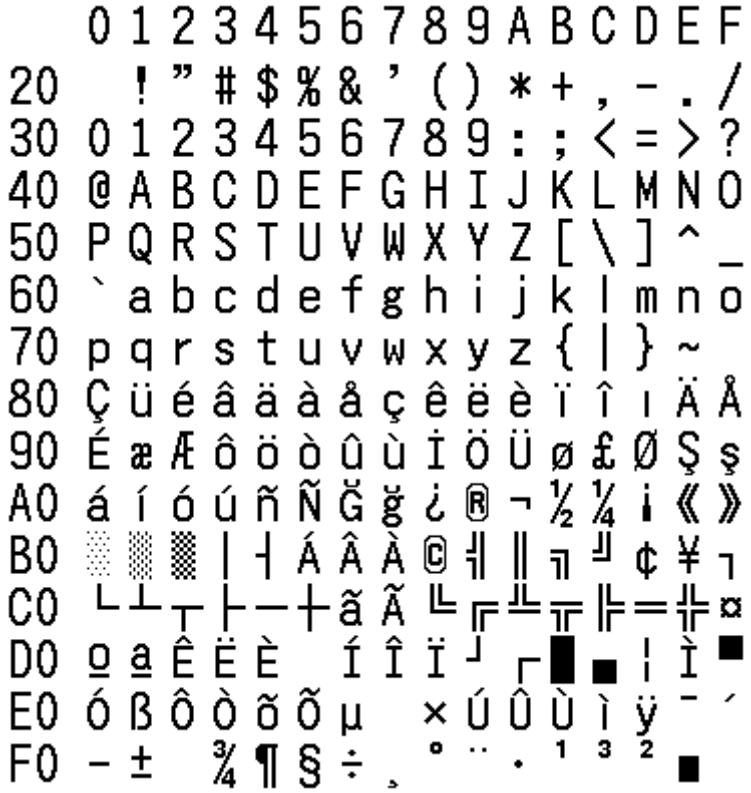


Figure A-7 SII_MPM_CODEPAGE_857 (Turkish)

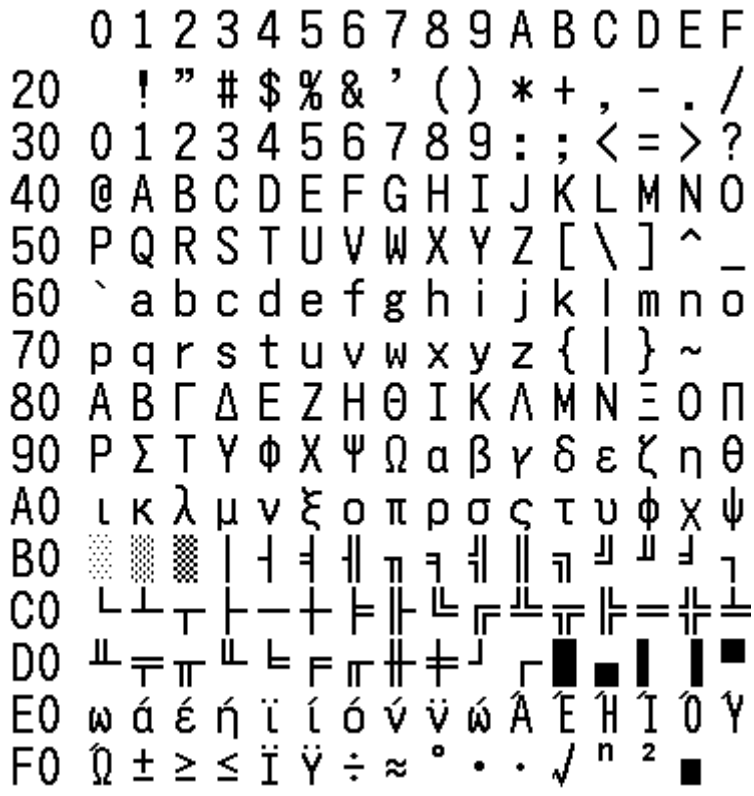


Figure A-8 SII_MPM_CODEPAGE_737 (Greek)



Figure A-9 SII_MPM_CODEPAGE_1252 (Latin)

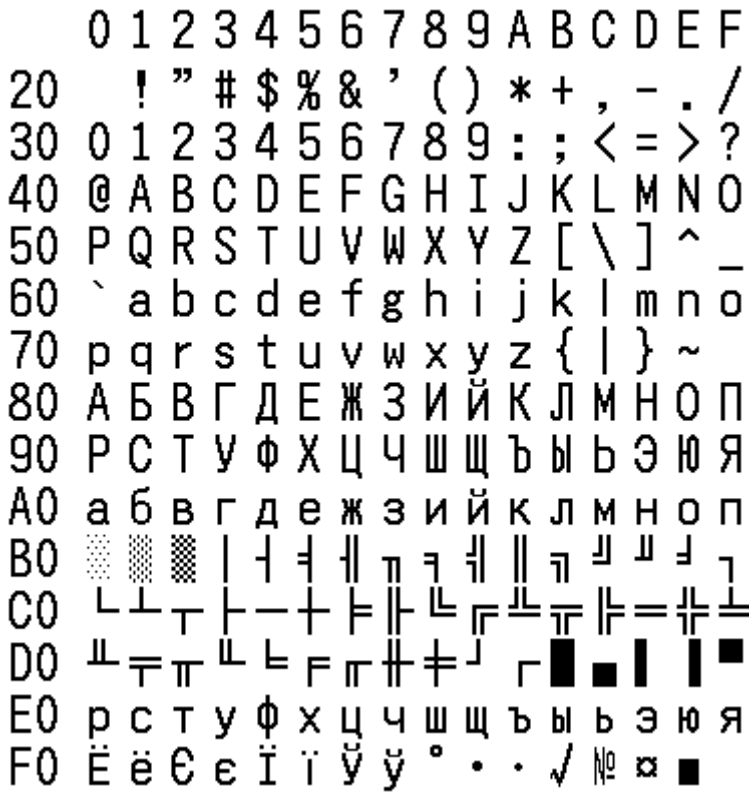


Figure A-10 SII_MPM_CODEPAGE_866 (Russian)

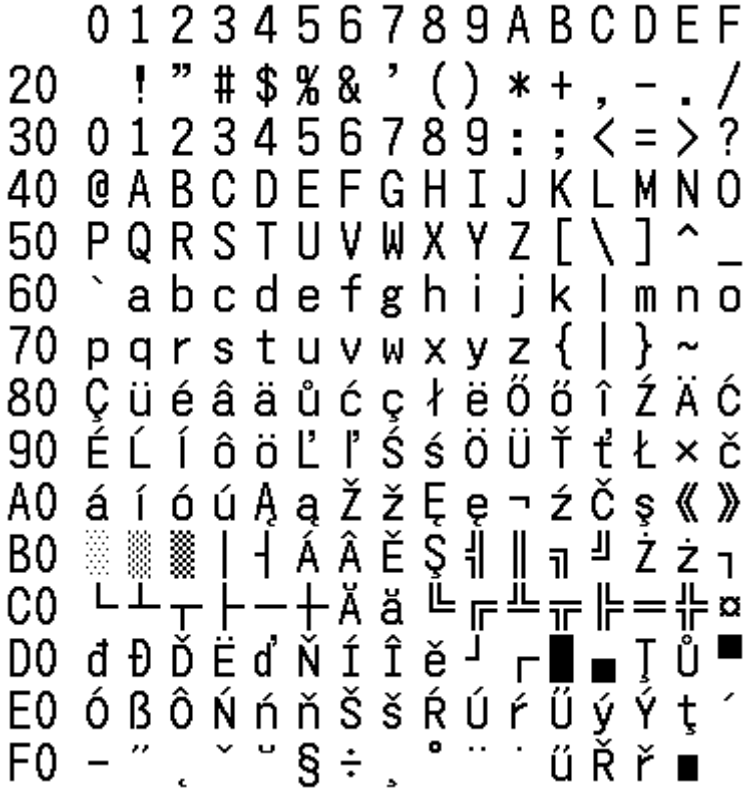
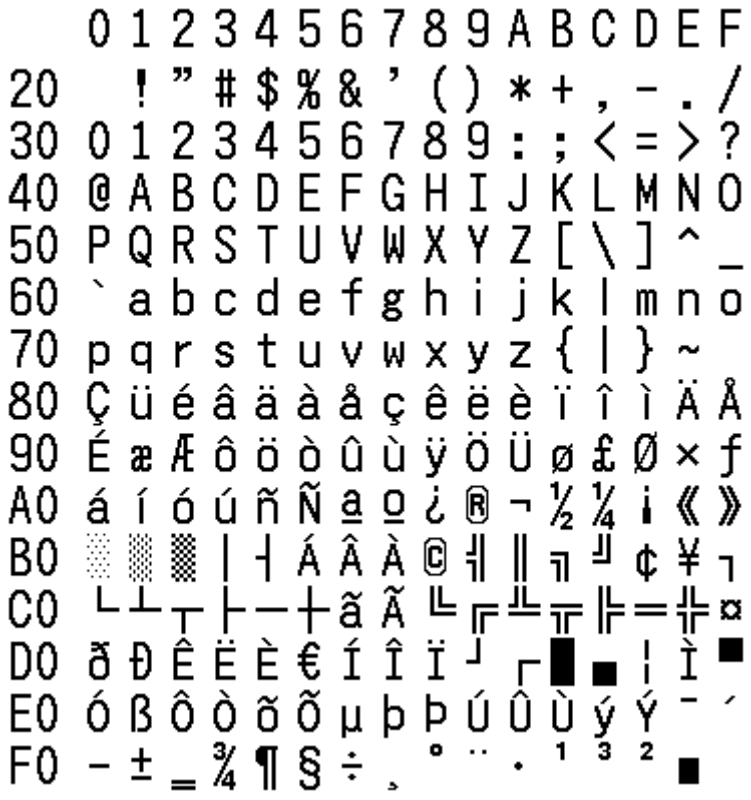


Figure A-11 SII_MPM_CODEPAGE_852 (Eastern Europe)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
B0	▒	▒	▒	|	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
C0	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	┘	┌	▒	▄	¦	Ì	▀
E0	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
F0	-	±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	▒	

Figure A-12 SII_MPM_CODEPAGE_858 (Euro)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	ђ	Ђ	ѓ	Ѓ	ё	Ё	є	Є	s	S	i	I	ї	Ї	j	J
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	ў	Ў	џ	Џ	ю	Ю	ъ	Ъ
A0	а	А	б	Б	ц	Ц	д	Д	е	Е	ф	Ф	г	Г	«	»
B0	▒	▒	▒	|	┤	х	Х	и	И	╣	║	╗	╝	й	Й	┐
C0	└	┴	┬	├	─	┼	к	К	╚	╔	╩	╦	╠	═	╬	¤
D0	л	Л	м	М	н	Н	о	О	п	┘	┌	▒	▄	П	я	▀
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	Ь	№
F0	-	ы	Ы	з	З	ш	Ш	э	Э	щ	Щ	ч	Ч	§	▒	

Figure A-13 SII_MPM_CODEPAGE_855 (Cyrillic)



Figure A-14 SII_MPM_CODEPAGE_864 (Arabic)

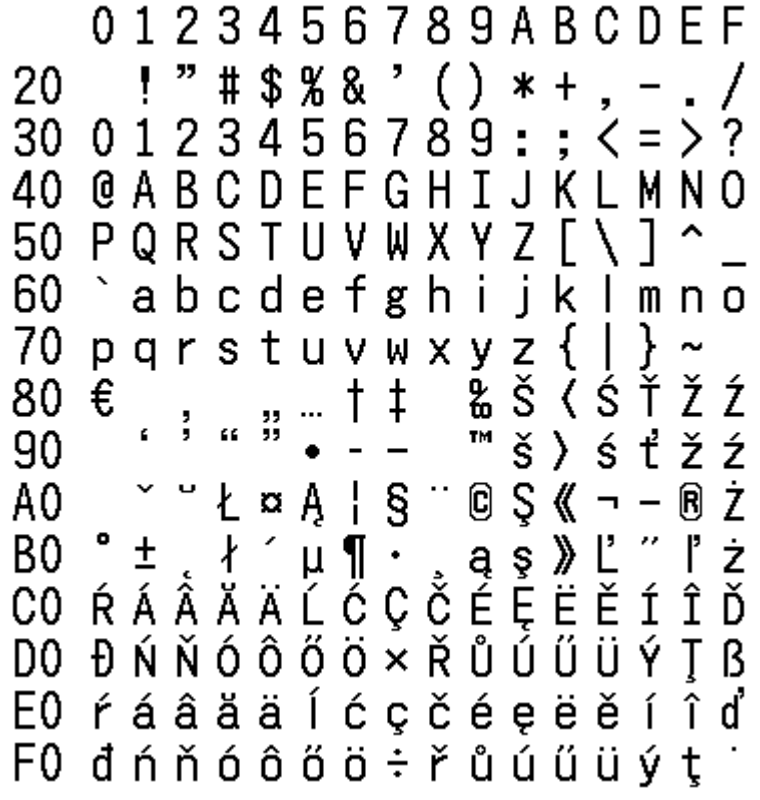


Figure A-15 SII_MPM_CODEPAGE_1250 (Central European)



Figure A-16 SII_MPM_CODEPAGE_1251 (Cyrillic)

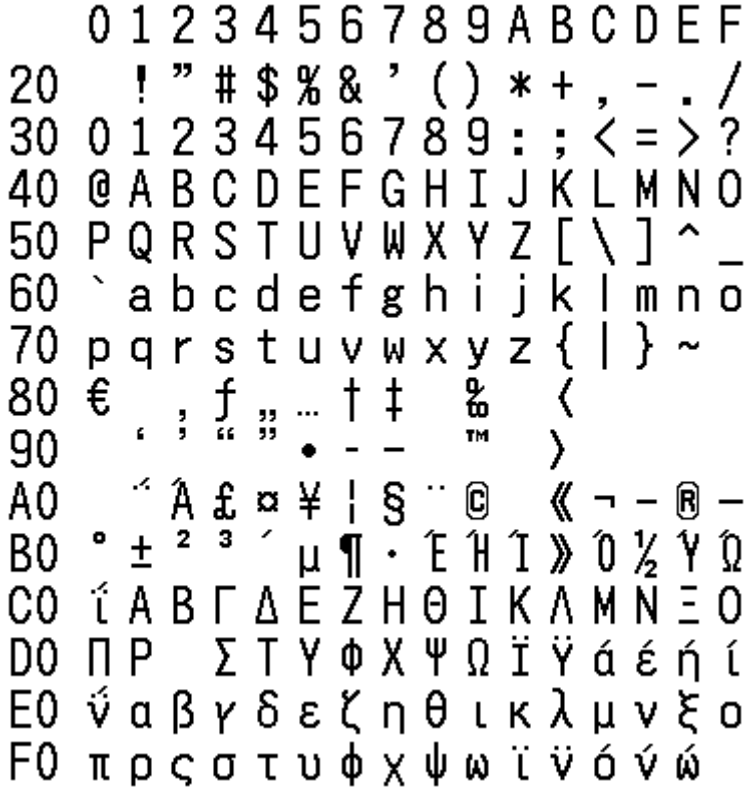


Figure A-17 SII_MPM_CODEPAGE_1253 (Greek)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	”	#	$	%	&	’	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
80	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š	‹	Œ			
90		‘	’	“	”	•	-	–	~	™	š	›	œ			Ÿ
A0		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

Figure A-18 SII_MPM_CODEPAGE_1254 (Turkish)

A.2 International Character Set

Print results of the specific character codes vary depending on the setting of the international character set. The following table shows the specific character codes and their print results.

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
COUNTRY_JAPAN	#	$	@	[	¥	]	^	`	{	|	}	~
COUNTRY_NORWAY	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	$	@	[	\	]	^	`	{	|	}	~

Figure A-19 International Character Set